Mastering EDL Test Points

By: Scott Lorenz

Foreword

Those who have been in the field for more than a few years have seen the dramatic changes in mobile forensic extraction techniques from the old Series 30 Nokia's to the latest smart devices today. Life has gone from impossible to easy, back to impossible, and swung over to a whole lot of maybe in the last ten years. The simple button pushing of the past has given way to deep research and a lot of frustration at times. Locked bootloaders, encryption, user passcodes, and other security measures have all conspired to make our lives more difficult.

As mobile operating systems have evolved, extraction techniques have fragmented into several different areas, from bootloader exploits, rooting, custom recoveries, chip-off, in-system programming (ISP), Joint Test Action Group (JTAG), unlocking exploits, and finally Emergency Download (EDL) techniques. It seems that the procedure for any phone today is "try stuff until something works." Luckily, we have someone whose motto seems to be, "I Try Stuff So You Don't Have To."

When Scott asked me to write a foreword for this guide, I originally thought that "ALL HAIL SCOTT!" repeated a hundred times would do the trick; I really cannot overstate the impact his research has had on the mobile forensic community. His untiring dedication to unlocking the secrets of phone extraction has no parallel; and the fact that he never sleeps helps. (I kid, Scott, I kid.) I remember taking a trip with him to a Fry's Electronics store after a JTAG class; he was running around, repeating, "I'll need this, and I might need this, and I need one of these, and…" Thankfully, that enthusiasm to dive deep into the innards of mobile phone technology has not waned over the years, as it has led to perhaps the greatest single-person contribution to modern mobile device forensics, and has contributed directly to ensuring that justice has been served in immeasurable ways.

This guide is the culmination (or perhaps just the beginning) of thousands of hours of Scott's dedicated research and development. I am certain we can all get invaluable insights from the contents, as so many have already.

Kim Thomson
Senior Examiner
H-11 Digital Forensics

Mastering EDL Test Points
By Scott Lorenz – Chief Forensic Analyst at Centex Technologies
06-14-2019
Edited by Andrew Rathbun

Document Purpose: This paper is the result of research into the existence of EDL test points on mobile devices, including myths and facts associated with EDL test points related to mobile device forensics. I specifically sought to locate EDL test points on individual devices and groups of devices made by various manufacturers including devices not known or believed to have EDL test points. In this document, I distinguish EDL test points from other locations and exploits often associated with EDL test points. I provide a definition for EDL test points that distinguishes them. I trace EDL test points back their origins - specific locations under Qualcomm processors. I identify and describe methods and techniques that can be used by examiners to locate test points on individual devices and groups of devices produced by specific manufacturers. I created and demonstrate the use of specific tools designed and adapted for finding test points, tracing test point paths, and creating EDL Mode on mobile devices. I provide detailed instructions on how to take full advantage of Cellebrite's use of EDL Mode to perform mobile device extractions. I hope you find the document informative and interesting.

General Overview: This document is designed primarily for the exploration of test points on mobile devices and includes information on exactly what test points are, where they are located, what they do, how to search for them, and how to use them to create EDL Mode. In covering test points, there will also be discussion on creating eMMC/UFS faults and the use of EDL cables to create EDL Mode. Other methods used to create EDL Mode will also be covered during the discussion of EDL test points. For general information of how to use EDL Mode to extract mobile devices using the UFED, refer to Mastering EDL Mode.

Requesting Access to Mobile Device Forensics and Analysis forum: When requesting access or membership in the forum, please include information about who you are (name), what agency you are from or your business or company, and your interests in the forum or purpose. That information in your initial request will save time and prevent forum moderators from having to send requests for more information.

Navigating This Document: First of all, Ctrl+F is your friend! Using a keyword search can quickly allow you to find specific items in this document. Each section of the document is a bookmark. Additionally, throughout this document there are links to photographs, diagrams, and other documents. Links may be in the text of the document or by clicking on a photograph or diagram. I have included some icons on photographs in this document designed to let the reader know if link leads to a photo or a video. It may be necessary to download this PDF for those links to work properly as opposed to just viewing it in your web browser. The links are to photos, videos, diagrams, and documents in the resources folder of the **Mobile Device Forensics and Analysis forum**. The Links in the Table of Contents lead to that corresponding section of the document.

Special Thanks: Thanks to Andrew Rathbun from HHS OIG for editing this document.

1 Overview of Mastering EDL Test Points

For those not familiar with EDL Mode and specifically Cellebrite's use of EDL Mode, I recommend you first read Shahar Tal's "*Practical Guide for Qualcomm EDL Physical Extractions*". That PDF document is also located in the Resources Folder/EDL Extractions, on the Mobile Device Forensics and Analysis forum. In that same folder is the PDF version of the "Cellebrite EDL Webinar 2-21-18". That will also give a general overview of EDL extractions – via Cellebrite's UFED. The 2nd Cellebrite EDL Webinar 9-12-18, is in PDF format. Mastering EDL Mode, is a detailed document that covers many aspects of EDL Mode and extractions using Cellebrite's UFED. In Mastering EDL Mode, there are links and references to many documents, diagrams and videos covering all types of extractions and procedures. Although Mastering EDL Test Points is designed to stand on its own, it was written to cover invasive methods for creating EDL Mode in detail and therefore this will not be a start to finish guide on EDL extractions.

1.1.1 The Need for Forensics

Although this paper is not designed as part of a philosophical discussion about the ethical use of mobile forensics, law enforcement practices, or research and development into breaking software and hardware to extract digital information, I know this document will likely be read by those who do not necessarily agree with any of these practices. To those who fear the use of these techniques you should understand that law enforcement and reputable forensic companies are bound by law and ethics. For the average person, it is very likely that your personal and private data will be accessed many times in your lifetime, but it will likely not be accessed using digital forensics. It is true with almost every political issue that we complain about things that are rare and unlikely while ignoring things that harm us every day. Your data will be stolen, leaked, traded, and sold by people and companies thousands of times with and without your permission and all of that will be done without having to conduct a forensic exam using techniques I describe in this paper. It is very unlikely that your personal or private data will ever be accessed by law enforcement or forensic companies using techniques and procedures discussed in this document. If they ever are, professionals provide an explanation and justification for why that took place.

1.1.2 Test points

Tests points can be found on many electronic devices, but this paper is confined to discussing test points on mobile devices designed to create EDL Mode. Test points were not installed on devices for the purpose of allowing forensic analysis. They were placed there to allow engineers and developers to test and troubleshoot the device's hardware and software. Just like any other exploit used by forensic examiners, test points can be used by forensic examiners to communicate with the device and extract digital evidence.

1.1.3 Reasons for researching test points

I will cover devices in this paper which have already been exploited via EDL Mode. I will also cover devices in which I have previously created videos and diagrams of eMMC fault locations. Test points are designed to be an alternative and, in many cases, can provide a better, easier, and safer way to get the device into EDL Mode. The location of test points on some devices means that EDL Mode can be created without removing heavy heat shields or removing and flipping the logic board. So, test points should ideally put the examiner in a better place regarding a forensic exam. Test points also provide an alternative method for creating EDL Mode where other methods do not work or are not available.

1.1.4 Detailed Device Diagrams

I attempt to provide as many diagrams as I can when putting together documents like this. I will generally include links to diagrams throughout the paper to illustrate points and provide a visual for what I am describing. I will also include a comprehensive list at the end of the document. It is very beneficial to be able to visualize what is being discussed regarding forensic procedures. A good diagram can help convey what I may not be able to clearly describe in writing. For some, the diagrams I create may be the only thing you look at regarding this paper and that is perfectly fine.

When I first started tearing apart phones, I very quickly realized the benefit of numerous photographs. In addition to the many diagrams I create, I keep the original photographs I take of each device in my inventory. I try to make diagrams as detailed and as clear as possible for several reasons. Clear, detailed diagrams are very beneficial in the understanding of how things work. Diagrams created for one purpose may also reveal other beneficial information not considered at the time the diagram was created. For some diagrams I may just include the location of test points, while other diagrams may include the

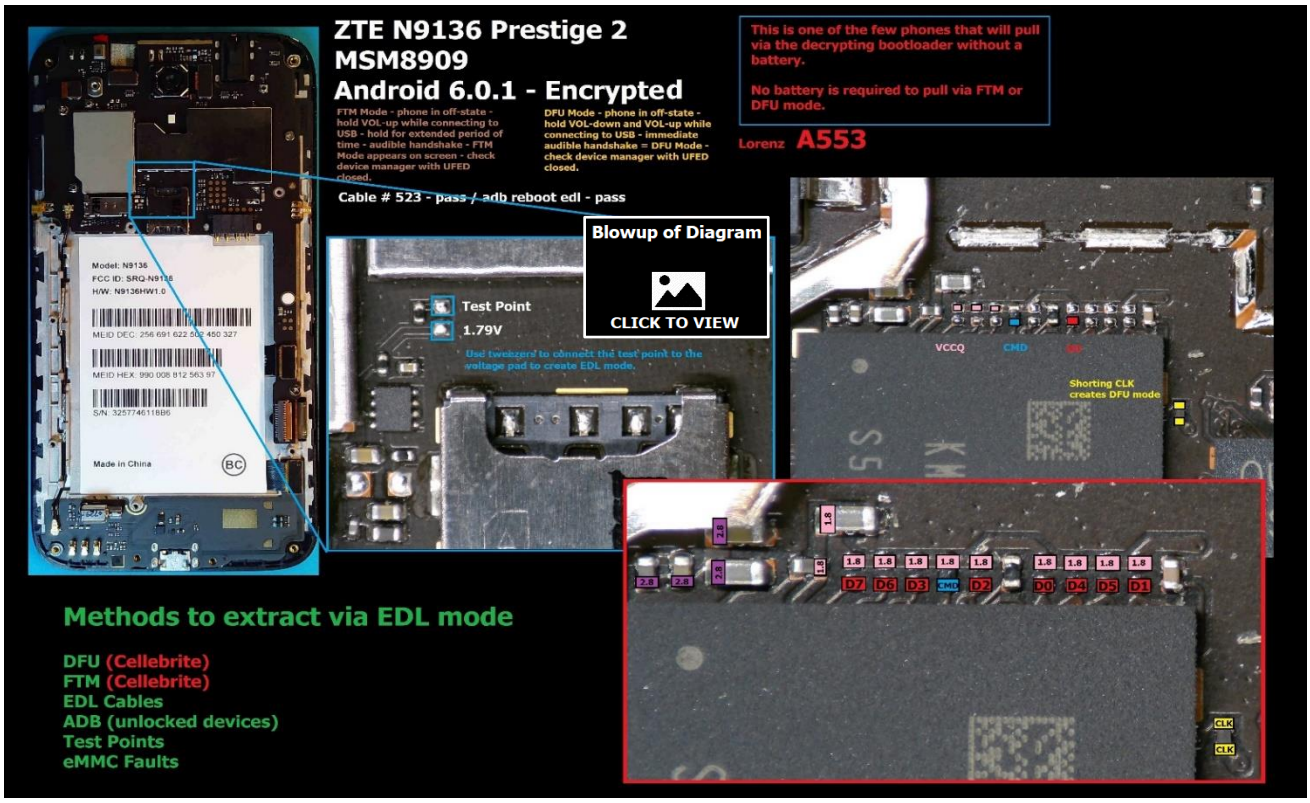
location of ISP points, test points, a pinout of the processor, and location of USB taps. That information will benefit me or someone else in the future.

When creating this document, I made several discoveries regarding test points and once I realized what I found, I went back looked at photos and diagrams of devices I had in my inventory. I realized that I had photographed an alternate method of extracting the device and didn't recognize it at the time I took the photo. I was able to identify the location of previously unknown test points on devices in my inventory just by looking at my old photos. In my opinion, there is no such thing as too many photos or too much detail. That was something I learned taking a JTAG class taught by Kim Thomson.

1.1.5 Diagrams with multiple ways to create EDL Mode

You will notice from some of my diagrams that I include several different methods for creating EDL Mode. So why would I mark test points or eMMC fault locations on a diagram when I know the EDL cable will create EDL Mode on that device? It is because sometimes the EDL cable will fail, or a variant of that same device, or a change in programming can affect what actions create EDL Mode.

Regarding my level of detail, I would like to be able to tell you that all my diagrams include every known method for placing that device in EDL Mode. Some do, but not all. You may see device diagrams I create that show the location of test points but mention nothing about being able to put that device in EDL Mode via an EDL cable. Don't read anything into me leaving that information off other than I may not have known or tested an EDL cable on that device at the time I located the EDL test points. At the time I create a diagram, I try to include and test for all known ways of creating EDL, but sometimes I do not and sometimes the condition of my test phone limits what I can test for at the time I make the diagram. The bottom line is that it doesn't hurt to test for yourself before disassembly.



2 Defining EDL Test Points

Do we really need a definition of EDL test points? The answer to that question depends on who is asking and under what circumstances. For users of test points just purely as, a means to an end – EDL Mode to extract a device – then the answer is not really. However, that assumes you are only going to rely on test points being identified for you by someone else with a diagram telling you where to place a pair of tweezers. If you want to locate test points for yourself then it is helpful and necessary to know how they work and what all test points have in common.

I teach an introductory class on Organized Crime at a local college. In one of the textbooks I utilize, the author begins by defining organized crime. Howard Abadinsky defines organized crime by listing eight attributes necessary to categorize and distinguish it from ordinary crime and terrorism. It becomes apparent when you look to investigate and prosecute organized crime that it is necessary to focus on the means and the ends. It is also apparent, when it comes to defining a particular event or thing, that defining something makes it unique. The thing that separates one thing from another thing that has a similar outcome can be the addition of something, the absence of something, or sometimes both. What distinguishes organized crime from ordinary crime is a group of people with a hierarchy of leadership. The only thing that distinguishes organized crime from terrorism is the absence of something – political goals. The outcomes of all types of crime may be the same from the perspective of the victim but it is the means and motive by which the crime was accomplished that changes how we find it, investigate it, and prosecute it. It is the "definition" that is most important and useful when looking for organized crime.

To find EDL test points it is important to define what a test point is, so we are all looking for the same thing. This serves not just an academic purpose but a practical one as well. By practical, I mean I will tell you things (attributes) about test points that make them unique from every other thing or method that creates EDL Mode and that distinguishes them from other points that do something else. It also means that when you focus on looking for those specific attributes, test points stand out and become easier to locate.

2.1 Lorenz's definition of an EDL test point

I debated with myself about where to place my definition of an EDL test point – at the beginning or end of this section. I decided to place it at the beginning and the remainder of the document will explain how I arrived at this definition. An **EDL Test Point** is:

> ***A single location on a mobile device, connected to the same specific location on the BGA of a Qualcomm processor which is utilized and shared across multiple models and OEMs, for the specific and singular purpose of creating EDL Mode after the user applies the predefined trigger***.

The trouble with attempting to create a single sentence to define a thing is that, in the absence of a perfect and absolute universe, the definition, or part of it, could prove to be inaccurate in some circumstances. This is probably why Howard Abadinsky elected to make a list of attributes to define organized crime instead of a sentence or paragraph used by other agencies and individuals. In any case, I am confident that the definition will prove true in most circumstances and even with the few exceptions that may exist, this definition will still lead you to the test point – if it is there.

With that general definition out of the way, in the remainder of this paper, I will describe why and how I came to those conclusions which make up the definition, while describing how and where to find EDL test points. In addition to my definition describing what a test point is, it is also exclusionary, meaning that when you apply it literally, it excludes eMMC fault locations (ISP points in which creating EDL Mode is not their primary purpose) and other locations on phones which can accidentally create EDL Mode like voltage spikes and damage. Applying a short to an ISP location is applying a trigger to create EDL Mode, but those locations serve another purpose and thus are excluded from my definition. The CMD, CLK, and Data lines are needed to exchange information with the processor and store data and thus have a primary purpose outside of creating EDL Mode when shorted.

My definition also excludes button combinations and the EDL cable, which come close to fitting but are excluded by the phrase, "…specific and singular purpose". Some button combinations can create EDL Mode, but it usually requires more than one button and those buttons also turn the volume up and down. The EDL cable technically applies a "trigger" to a single

location, the D+ line on a USB cable, but that location doesn't have singular purpose. In fact, its primary purpose is to transfer data between the mobile device and a computer or forensic tool.

You may notice that my definition says nothing about the size and shape of test points. What about those little round pads we see all over the internet? You will find that many test points are little round pads, but as you will see from this paper and examples, you can't rely on just looking for little round pads. You will be surprised about where some test points are found and what the actual test point on some phones is. The physical shape and size of test points is important, especially on some OEMs, but that is only a small part of identifying test points. You will see that those "little round pads" are all over many devices. Only one of them is the EDL test point.

There is a practical consideration for all this discussion of a definition which goes to the core of why we should pick test points over eMMC faults when button combinations and EDL cables fail. Devices can't function normally without Data, CMD, and CLK lines or locations. It is therefore safer to use EDL test points than eMMC faults. In this paper, I discuss the issues associated with examiners damaging phones with procedures designed to produce eMMC faults to create EDL and accidentally damage the device so that it can't be extracted and/or booted. If while creating EDL Mode, you damage the Data line, CMD, or CLK line or resistors, you may not be able to boot and extract the device.

2.2 What are EDL Test Points?

With my definition of test points given, the details and methodology of how I arrived at that definition are important in learning how to find them. To better understand what EDL test points are, it is necessary to distinguish them from what they are not. In a very simple definition, EDL Test Points are a "means" of creating a desired outcome: "EDL Mode". In Mastering EDL Mode, I discuss the different methods to create EDL Mode.

Shahar Tal from Cellebrite must be given credit for this list of things (techniques) that create EDL Mode. Read his Practical Guide for Qualcomm EDL Physical Extractions. Some of the methods on the list I have below are unique to Cellebrite. In other words, they are methods that Cellebrite pioneered and/or are unique to Cellebrite EDL extractions. It is important to distinguish between techniques deployed by the examiner that create EDL Mode versus techniques deployed by the examiner that create DFU, FTM, or LAF, all which Cellebrite's UFED can convert to EDL Mode. This means Cellebrite may be able to take advantage of devices (create EDL Mode) in DFU, FTM or LAF while other forensic software may not be able to or may not offer that exploit.

1. Key combinations – e.g. holding Vol Up and Vol Down (locked and unlocked devices)
2. Cable 523 (**Cellebrite**) or homemade EDL cables (locked and unlocked devices)
3. ADB 'adb reboot edl' - (unlocked devices)
4. Fastboot 'fastboot oem edl' (unlocked devices)
5. DFU and FTM Mode - (**Cellebrite**) (locked and unlocked devices)
6. LG's Advanced Flash - (**Cellebrite**) (locked and unlocked devices)
7. Test points (locked and unlocked devices)
8. eMMC fault injection (shorting) (locked and unlocked devices)

From the list above, you see that Test Points are in their own category. I have them listed at #7, on the list because generally I list methods of creating EDL in terms of levels of invasiveness to the physical device. The list is similar to a use of force continuum, to draw a metaphor from law enforcement terminology. Test points and eMMC fault injection, #7 and #8 respectively, require device disassembly. For a review of all of these procedures, please refer to my Mastering EDL Mode guide. If you follow typical procedure, utilizing EDL test points comes after other less invasive measures have failed or are otherwise not available – a common occurrence. Just like the law enforcement use of force continuum, there may be reasons for skipping the other options, which may be available in the continuum, and I also discuss those considerations in Mastering EDL Mode.

For some devices that are specifically supported for an EDL extraction in the UFED under that specific device profile, Cellebrite will include instructions for how to prepare the device for extraction. The instructions may include button combinations. Those button combinations do not always place the device directly in EDL Mode. Many of those combinations

create DFU, FTM, or LAF Mode but you may not always know what specifically is occurring. It is not necessary for the examiner to know what is being created when operating under the UFED's device profile because following the instructions will normally lead to an extraction. However, when using generic options in the UFED or when using another forensic tool, it is necessary to know what the button combinations do. If you mistakenly believe your device is in EDL Mode but it is in DFU Mode, an attempted extraction with some forensic tools may fail.

2.3 EDL Mode is a creation of Qualcomm

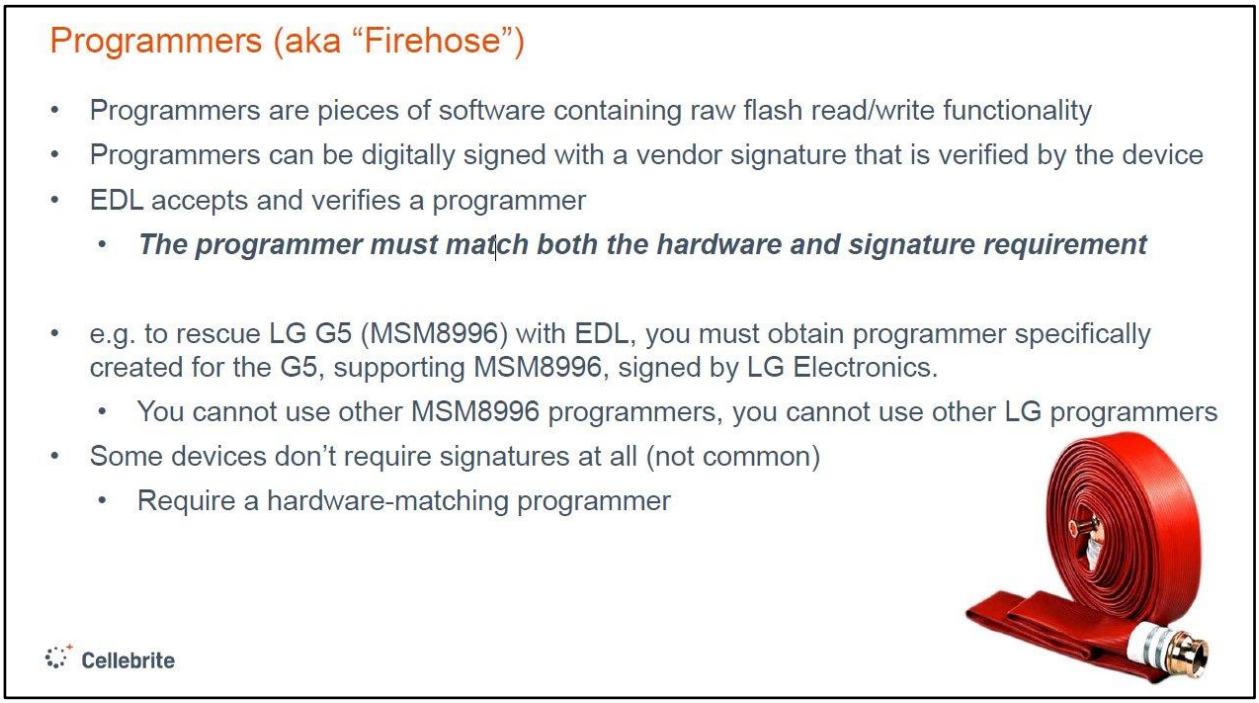
EDL Mode is a creation of Qualcomm and test points installed on hardware are generally placed there to allow developers and programmers an ability to create EDL Mode for testing, analysis, and repair. So EDL Mode is not a creation of forensic companies. Like many other methods used by forensic examiners, EDL Mode is used to access and extract information on a device by taking advantage of the device in a vulnerable state. Just because you find a way place a device in EDL Mode, does not mean an extraction will follow. Without a firehose bootloader for that specific device or an exploit capable of taking advantage of most devices running a specific processor, creating EDL Mode might not result in a data extraction. In other words, you may be able to place a device in EDL Mode, but the device is not supported for an EDL extraction.

2.4 EDL Mode doesn't mean you can extract the device via the EDL exploit

In this paper, I provide diagrams and methods which will place many devices into EDL Mode. I provide maps and diagrams to create EDL Mode for devices that are currently unsupported for an EDL extraction. I do that because someday there may be Firehose programmer for that device or there may be another exploit related to EDL Mode for that device in the future. Getting a device in EDL Mode is only a first step. Do not assume an extraction will always follow.

2.4.1 Firehose programmers

Shahar Tal from Cellebrite described the process needed to extract devices in EDL mode in the 2nd Cellebrite EDL Webinar 9-12-18. Digitally signed programmers are needed to be able to extract data from the device via EDL. Those programmers are digitally signed by device manufacturers. Some of those digitally signed firehose loaders have been made available publicly but many have not. Thus, when an extraction failure occurs after a device is in EDL mode, the usual reason for the failure is the lack of the digitally signed firehose programmer.



2.4.2 Cellebrite's ability to bypass digital signature checks - "Widely Supported" chipsets

Tal also describes Cellebrite's proprietary technique to bypass signature checks for five chipsets described by Cellebrite as "Widely Supported". Cellebrite uses a proprietary technique to bypass the check for a digital signature for five Qualcomm chipsets (MSM8909, MSM8916, MSM8936, MSM8939, and MSM8952). Devices running these chipsets will nearly always extract under Generic options the UFED, even if they are not listed as supported for an EDL extraction under their device profile. These processors, especially the MSM8909 and MSM8916, have been used on many different models, including models not listed at all in the UFED. This is one of the reasons for becoming familiar with using the Generic options in the UFED.

2.4.3 Programmers for specific devices "Limited Support" and other unknown support

The UFED contains a library of Firehose programmers which may or may not work on devices running Qualcomm processors like the MSM8917, MSM8937, MSM8940, MSM8953 and MSM8996. There are also some devices running other Qualcomm processors which may extract via EDL under Generic options in the UFED. This may be because of the very few devices that do not require digitally signed programmers or it may be due to shared characteristics of some Qualcomm processors not on the widely supported or limited support list. The SDM450 is very similar to the MSM8953. There are also some variations of MSM8953 processor in which one 8953 may be different from another 8953, even in terms of BGA pin assignments. There are some devices running the SDM450 which will extract via EDL. Some devices running the MSM8929 will extract via the EDL exploit in the UFED, (e.g. Alcatel X1 – TCL X1 – 7053E – MSM8929). The MSM8929 is very similar to the MSM8916. Attempting EDL extractions under Generic options in the UFED is harmless to a device, even if the device fails to extract.

2.5 Typical EDL Test Points

The best way to start the discussion of EDL test points is to cover and look at what most examiners already know. I will provide some examples of common test points that most of you will recognize. There may be things about test points of which many people are not aware. Understanding some details about test points will help examiners better understand how to use them and how to find them.

2.5.1 Disclaimer to the rules

I frequently emphasize the need to never say "never" or "always" when referring to any procedure or pattern in digital forensics or generally anything else in the universe. Please keep in mind that I am discussing phones common to the United States and manufacturers with which I most frequently examine. I generally believe these guidelines are consistent worldwide, but I have not had the chance to test every device – not even close.

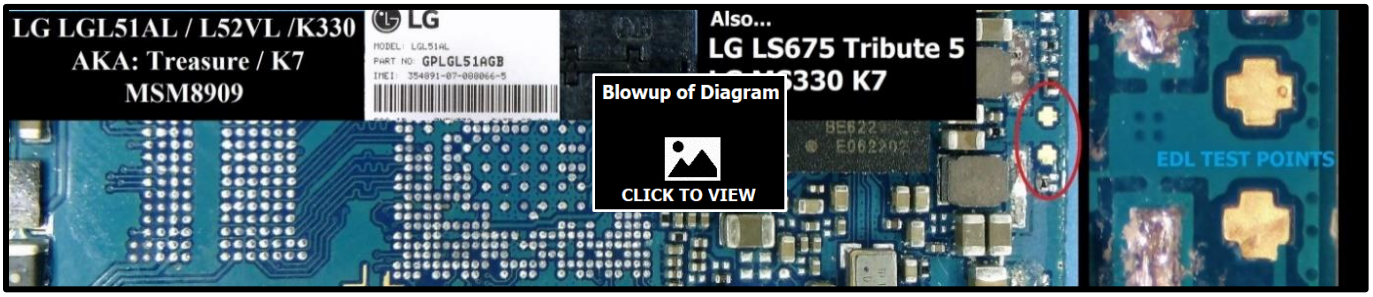
Before I write any document and release it to be viewed, I have not tested all the phones I would like to test. In fact, I continue testing the procedures and patterns I have already published after I publish my findings. No forensic company in the world has been able to test every device that hits the market. It is simply a matter of physics. There are too many devices being release too quickly. My R&D team consists of just me. I test what I can and what comes across my path through many different sources and means. I use only diagrams I create myself and have pinned or tested myself because I want to see things for myself. Don't get me wrong, I rely on information from everyone else's experiences on forums and contacts I have with people in the business as well. Therefore, the disclaimer is, there are exceptions to these rules and I don't know what I don't know.

2.5.1.1 Assumptions made across device variants

For anyone who has worked with mobile forensics for any significant amount of time, you realize that forensics companies and examiners must make assumptions about device support and behavior. In other words, predictions are made about devices we have not held in our hand based on similar devices we have tested. Identifying a device can sometimes be the cause of great confusion and frustration and is challenging even for large companies who specialize in forensics. In this document, I will include a list of devices and diagrams identifying test points on devices I have tested and on devices I have not tested or even held in my hand. When I locate an EDL test point and confirm that EDL Mode is created on a device, I may apply that diagram to several variants of the device I tested. I check FCC IDs, databases, forensic software, device reviews, and information from other examiners. It is important to understand I have not actually tested all of the variants, which means I am making an assumption based on all of the information I have.

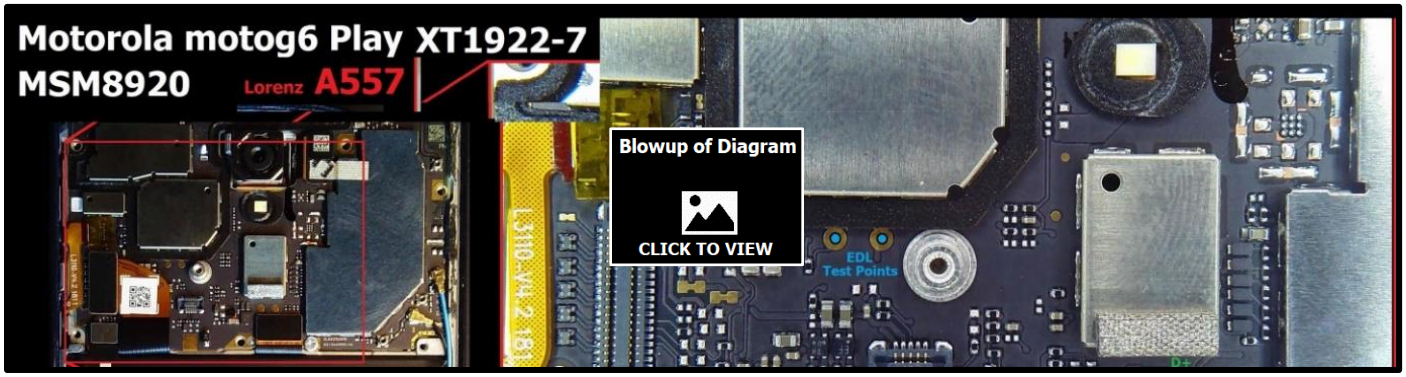
2.5.2 EDL test points on LG phones

On many devices, EDL test points stand out or are distinguished by location, proximity to each other and, with LG devices, the shape of the test point. You can see in the diagram of the LGL51AL that LG made sure their EDL test points stand apart from anything else on the phone. You can also see that the test points are located on the outside of the heat shield which covers the processor and storage. Thus, "access" is one of the reasons I list test points below eMMC faults on level of invasiveness. LG test points are always easy to spot, even if you are only looking at FCC photos. They are small but unmistakable if you know what you are looking for.



2.5.3 Typical Motorola test points

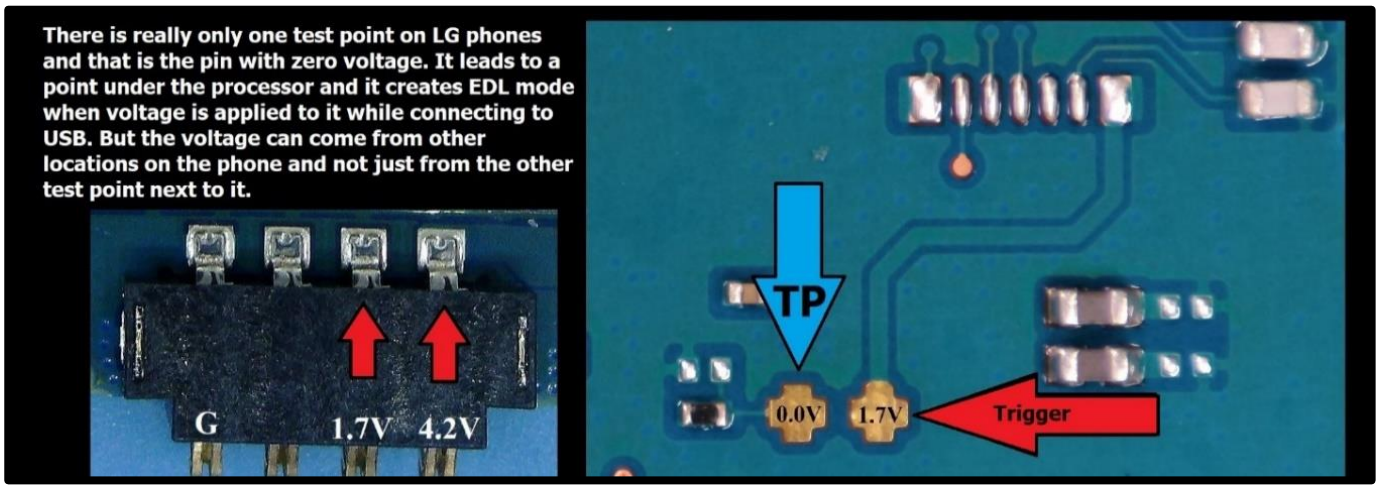
Motorola is also known for having EDL test points on many phones, but they do not make them as easy to spot as LG test points. Motorola is like many other manufacturers who use round pads for EDL test points. The pads are not hard to spot, but that is not the problem. The problem with EDL test points that are round pads is that there are many round pads on many different phones. So which ones are the test points? Some phones have many round pads and none of them are EDL test points. Still a good way to start the search is by looking for the obvious first.



The diagram of the Motorola motog6 Play XT1922-7 provides an example of round EDL test points. Notice that there are many round pads visible in the diagram and some of them I have identified as other things like USB pads. When I first went looking for EDL test points on this device the two EDL test points I have marked are what I tried first. Their proximity to each other, location on the top side of the logic board, and the fact that I know what some of the other pads were, made those two pads stand out. This device is still not supported for an EDL extraction as of the writing of this paper. It is important to remember that most all modern devices running Qualcomm processors can be forced into EDL Mode, but that doesn't mean there is an available exploit to take advantage of that device. So just like the LG EDL test points, the two pads on Motorola phones are usually located next to each other in a convenient position. Place some tweezers on the pads, plug the phone into USB and bam, EDL Mode. If the device is supported and if we have done things in the correct order with whatever exploit we are using, a memory dump is not far away.

The myths about EDL test points

Using the term “test points” is technically incorrect when you consider what occurs with a pair of tweezers. Not understanding what EDL test points are and what they do has made them more difficult to locate in some instances and led many to believe, myself included, that some phones just didn’t have EDL test points.



2.6 There is only one test point

When test “points” don’t exist or are not known, examiners go to the last option which is eMMC faults (shorting). Creating eMMC or UFS faults is not the same as using a test point to create EDL Mode. Even though the ends (EDL Mode) is the same, the means of creating EDL Mode via faults versus a test point is different. There is only one EDL test point.

Therefore, of the two pads used to create EDL Mode, only one of them is the “test point”. The other one is what I refer to as the “**trigger**”. EDL Mode is caused by connecting the trigger to the test point and then applying power to the device. Connecting the trigger to the test point is usually accomplished by a pair of tweezers because the trigger is most commonly and conveniently located next to the test point. The trigger is not unique. It is voltage, which can come from anywhere. As always, there are some exceptions with some devices.

Does it really matter? There is no wrong way to use tweezers. Continuity is continuity, right? That is true if you assume that all test points look alike and if you assume they are always located next to their trigger. Before going further with this explanation and going into detail with each myth, I need to name the other myth about EDL test points as it will be impossible to discuss one at a time without stepping on the other.

2.7 Neither the test point nor the trigger is a ground

A common assumption about test points is that EDL Mode is created by grounding something. When using an actual test point, EDL Mode is created by connecting the trigger (normally 1.8 volts on modern devices) to the test point, which is normally no voltage at all but is not ground. The exception to that rule is Alcatel, in which the test point is voltage and the trigger is ground. Another reason for the belief that test points involve grounding something is due to the use of the word “testpoint” to describe ISP locations. The diagram of the LG D325, from Octoplus is an example in which the CMD location is described as a test point. That diagram is labeled “testpoint”, but it marks the location of CMD.

Figure 1. Testpoint for LG D325.

Please don't misinterpret any of this to mean I am being critical of the practice of identifying anything that creates EDL Mode as a test point. In the absence of knowing what ISP location you are using and for the purpose of using a term most people understand, "testpoint" is probably the best way to keep the diagram simple. For the purpose of this paper, it is necessary for me to distinguish test points from other locations on phones that create EDL Mode. Storage faults (eMMC or UFS) involve applying a ground to ISP points. Thus, we use the term "shorting" when referring to the creation of storage faults. Creating those faults involves applying a ground to Data, CLK, and CMD locations.

2.8 Modern vs. older devices – test point characteristics

EDL test points have changed over time. For example, there are EDL test points on older LG devices, but they are not two very distinct crosses located next to each other. The voltage for modern test points (no voltage for the test point and 1.8 volts for the trigger) is also not always the same when you go back in time looking for test points. Test points I located on the Galaxy S4 and S4 Active tested for about 0.7 volts and the trigger was almost identical voltage. Therefore, when I refer to common patterns and voltage readouts of test points and triggers in this paper, I am speaking about more contemporary devices unless I make a distinction. It doesn't mean there are no similarities, but there are differences and OEMs have changed over the years. I did go back in time with some research, but the amount of time involved, and the sheer number of devices is staggering. My work on older devices is only preliminary.

Test points are not ISP locations

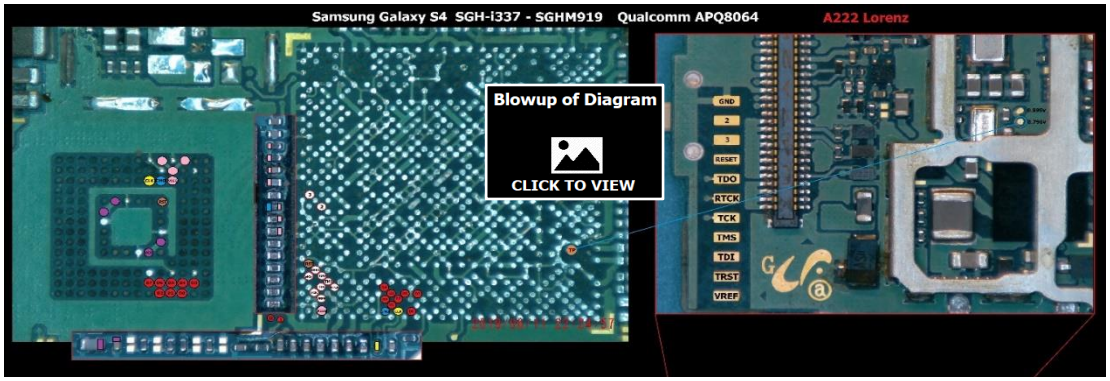
ISP (In-System Programming) locations are not EDL test points. Many of the flasher boxes refer to ISP locations as a "testpoint" in their diagrams. If the definition of a "testpoint" is something that creates EDL Mode when you apply a ground to it, then Data, CLK, and CMD are all test points. The problem with using that definition to go looking for hidden test points that we have all missed, is that none of those locations will lead you to actual EDL test points. Soldering into an EDL test point will not help you ISP a phone as actual EDL test points are not Data0-7, CMD, or CLK. The EDL test points we are all familiar with, like the crosses on modern LG devices, have nothing to do with ISP as grounding them will do nothing.

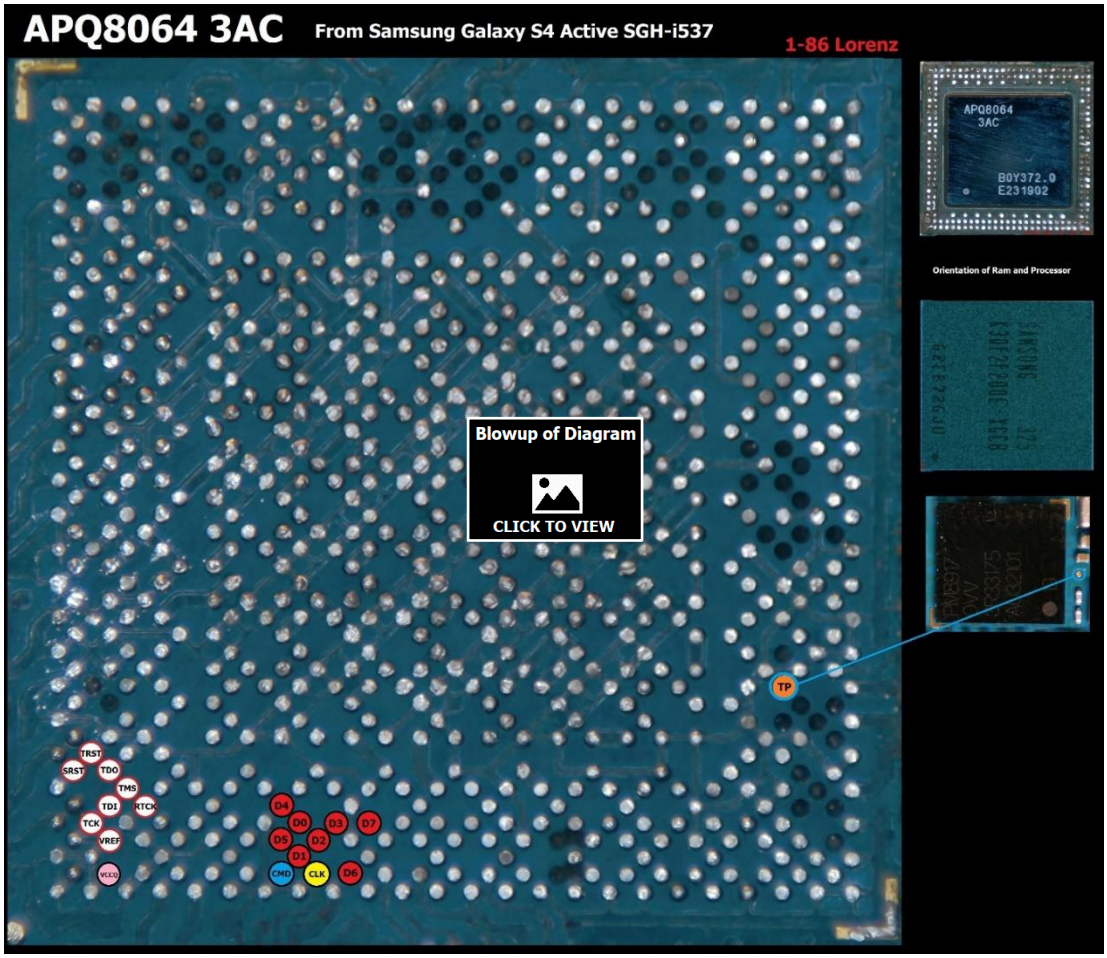
2.9 Test points are not JTAG TAPS

JTAG TAPS (Test Action Ports) are also not EDL test points. Not only will JTAG TAPS not lead you to an EDL test point, but grounding or applying voltage to JTAG TAPS will not create EDL Mode like grounding and applying voltage to ISP locations will. Grounding an actual EDL test point will not create EDL Mode on most devices (exception being Alcatel). EDL test points require voltage to trigger EDL Mode. When you find an actual EDL test point on a phone with known JTAG TAPS, you will find that the EDL test point has nothing to do with NRST, RTCK, SRST, TCK, TDI, TDO, TMS or TRST. There are known JTAG TAPS on the Samsung Galaxy S4, SGH-i337. The phone is running eMMC storage, so it can be exploited via ISP. I went back and looked at this phone for research on this paper and I was able to locate an EDL test point and trigger under the heat shield and modular microSD card and SIM card slot. Those points reliably create EDL Mode and I was able to distinguish them from the JTAG TAPS and ISP locations. I was able to create EDL Mode via eMMC faults (shorting exposed ISP locations – CMD, CLK, D0 & D1). I tried to create EDL Mode using the JTAG taps to including shorting, applying voltage, and connecting them to each other with tweezers. I could not create EDL Mode with any JTAG TAPS.

The actual proof that EDL test points are not ISP locations or JTAG TAPs can be demonstrated with the use of a multimeter. ISP locations, JTAG TAPs, and EDL test points can be traced back to the Qualcomm processor. JTAG TAPS and ISP locations are

not connected to each other or the test point and all lead to separate pins on the Qualcomm BGA. ISP locations can be used to create EDL Mode and may even be more reliable in some circumstances, but they are not EDL test points. JTAG TAPS do not create EDL Mode when applying ground or voltage. This can be observed by examining a phone with a Qualcomm processor, supported for ISP, with known JTAG TAPs. I located the EDL test point on those phones and confirmed the test point was not connected to JTAG or ISP. Once locating the test point on the BGA of the APQ8064, I was able to trace that to the location of test points on other devices running that same processor.

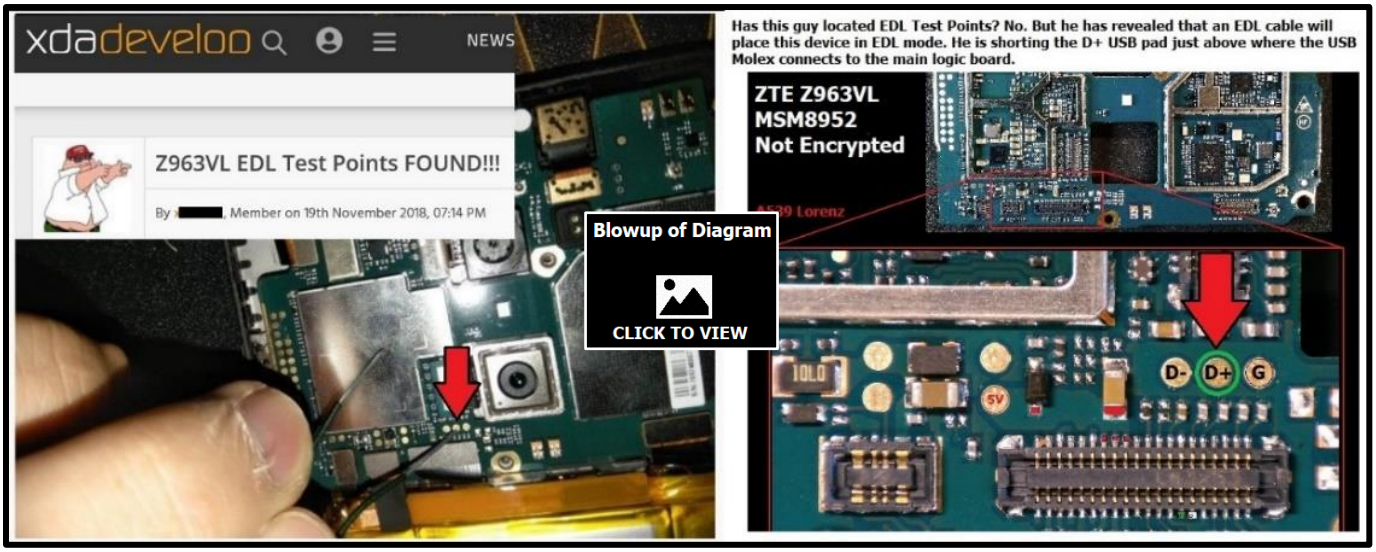




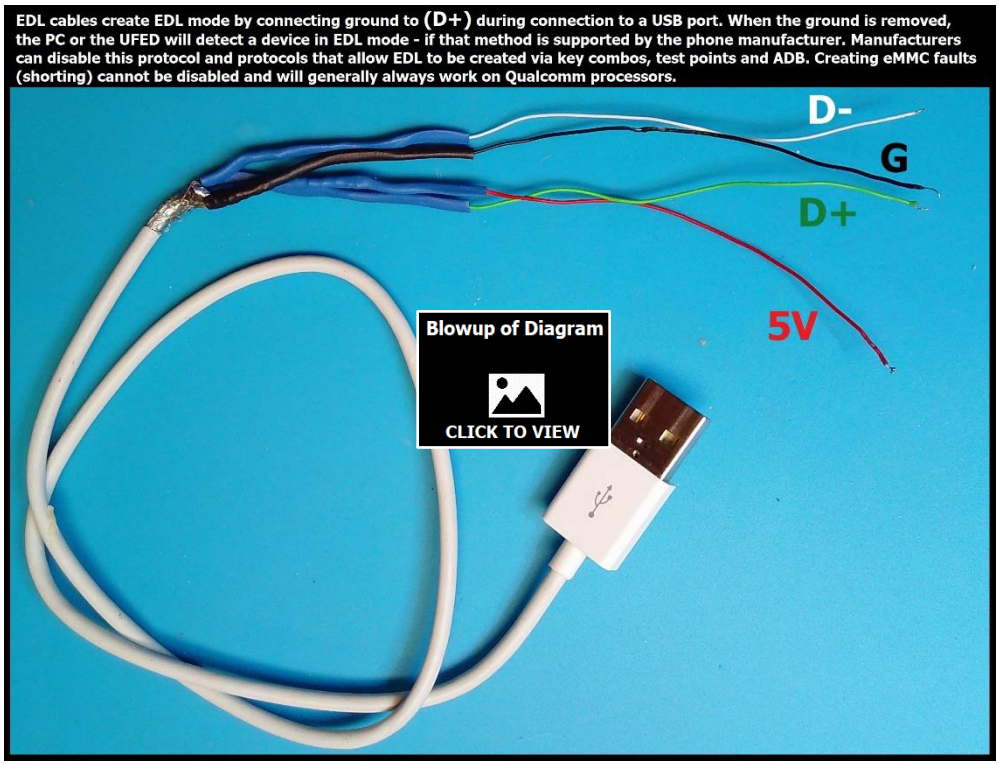
2.10 Confusing the USB D+ pad for a test point

Knowing precisely what is taking place when using a technique to create EDL can be useful when looking at diagrams or searching for easier methods to extract phones. For example, knowing test points on ZTE phones involves applying voltage to a location, combined with the knowledge that EDL cables create EDL Mode by applying a ground to the green D+ line inside a USB cable, can help us find a simple way to get the Z963VL into EDL Mode.

Looking at the photo of the Z963VL posted on XDA can tell us that an EDL cable will work to extract this device even if we didn't have the device to test or didn't already know. Why? Because the test point on ZTE phones is triggered by voltage, not ground. In the XDA photo, one side of his tweezer is on ground (the heat shield) and the other tweezer is on a round pad that looks like a test point but is located directly above the USB Molex connected to the main logic board. Though he calls it a test point, he is grounding the D+ pad. With that photo, I know I can create EDL Mode without taking this phone apart. What he is doing can normally be accomplished with an EDL cable (e.g. Cellebrite's cable # 523).

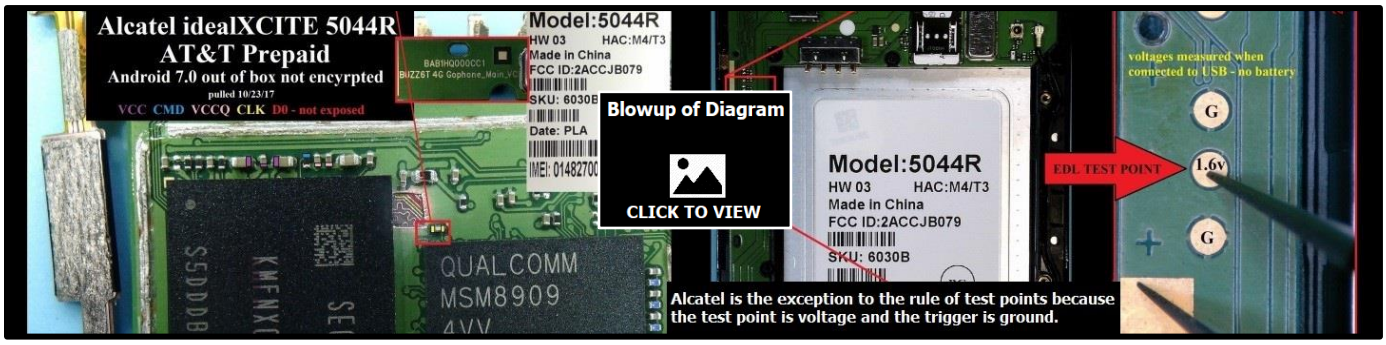


So why not use cable # 523 on everything? EDL cables won't work on every phone and if a properly used EDL cable wouldn't place this device in EDL Mode, the method shown in the xda-developers forum photo won't work either. On devices in which an EDL cable fails, grounding the D+ pad on the logic board will also fail to create EDL Mode. Some OEM's can and do shut off the ability to create EDL Mode via EDL cables.



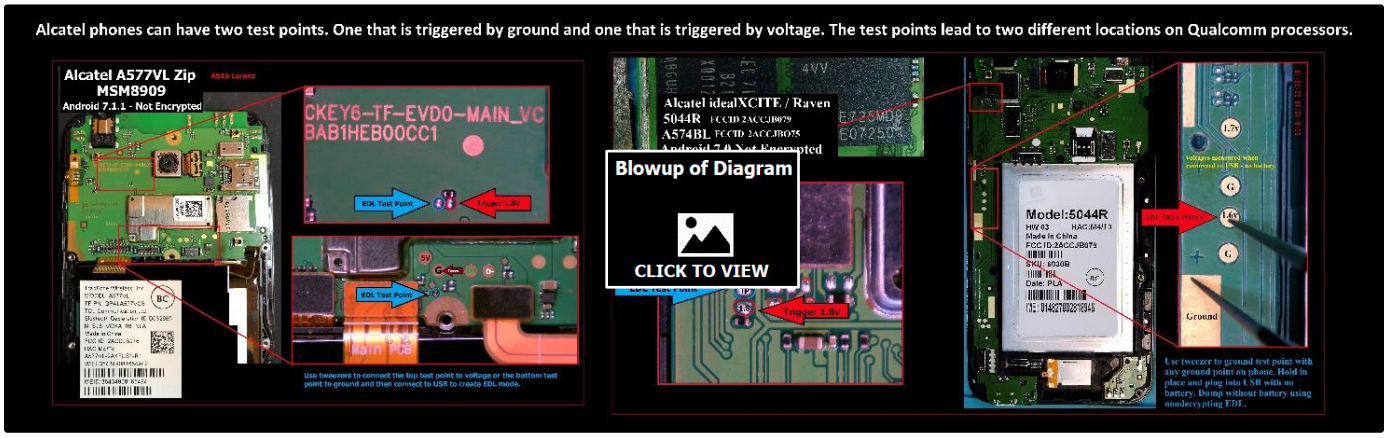
2.10.1 Alcatel is the exception to the rule that the trigger is voltage

Of course, it wouldn't be digital forensics if there wasn't an exception to every rule. When it comes to the familiar round pads on phones, the trigger for the Alcatel test point is ground. An example of these familiar pads can be seen in the photo of the Alcatel 5044R. Those round pads are what is recognized on most Alcatel phones and many other phones. Therefore, Alcatel behaves exactly like the widely believed myth that everything involves grounding, but Alcatel is the exception to the most



common rule that test points are triggered by voltage. Another difference with Alcatel phones is that the actual test point will usually test for 1.5-1.8 volts as opposed to no voltage on most other devices. Of course, that makes sense when you think about it. A trigger requires a change in something normal. If the Alcatel test point had no voltage like and LG test point, grounding it would do nothing. It is also why the trigger for a test point which operates in a no-voltage state, needs voltage to trigger it. The test point on most other devices running Qualcomm processors is most often, no voltage or very low voltage, requiring a trigger that is voltage. There are exceptions to that rule on some devices, which involve a trigger next to a test point and both test for 1.8 volts. But wait, there's more with Alcatel…

The most recognizable test points and triggers on Alcatel phones is the round pads. The test point is voltage and applying ground causes the phone to default to EDL Mode. But is that the only test point on Alcatel phones? No. Some Alcatel phones have traditional test points as found on other phones – that is the test point is no voltage and the trigger is 1.8 volts. These test points on Alcatel phones are in a different location and are not as obvious as the familiar round pads, but these are test points. On Qualcomm processors, there are two test point locations I have identified with Alcatel. One of them is unique to Alcatel and the other test point location shares the same processor BGA pin as other OEMs. So, Alcatel breaks another rule. There is only one test point on a phone…except some Alcatel phones that have two.



2.11 The general rules of test points encapsulated

It is helpful sometimes to start with broad rules and then narrow to specifics. The following is a list of general concepts. The specifics of locating and using test points and details related to these general rules will be covered later. Please remember that nothing is absolute when it comes to mobile forensics. I try to list the exceptions, but changes occur frequently so consider everything here as guidelines that are generally correct.

1. **Test points are unique** – usually only one on a phone. Exceptions (Alcatel). Remember this is under my definition of a test point which doesn't include eMMC or UFS fault locations, JTAG TAPS, the D+ line or pad, or voltage irregularities which can force a device into EDL Mode.
2. **Test points are triggered by voltage** – Exceptions: Alcatel's most recognizable test points are triggered by ground, but some Alcatel phones also have a test point triggered by voltage which leads back to the same BGA location under Qualcomm processors – the same as test points on other OEMs.
3. **The trigger is 1.8 volts** – On many devices, the trigger is normally 1.8 volts (precise readout may be 1.79v) but variations occur, and some devices require much more voltage to be applied to a test point to trigger EDL. I will give examples involving Google and ZTE.
4. **The trigger is near the test point** – Most of the time this is true but there are exceptions, which usually means more than 1.8 volts will be required but not always. There is also not always a clearly defined ground pad associated with a test point on some Alcatel phones. I have identified several Alcatel phones that place a pad linked directly to CMD on the top side of the board very conspicuously. Because Alcatel test points like that are triggered by ground, I mistook it for a test point until I traced it to CMD. It works the same either way – ground it and you will get EDL.
5. **Test points are conveniently located** – Usually true but everything is relative. By conveniently located I mean outside of heat shields and on the face-up side of the logic board so that it doesn't have to be removed. However, some are located near the processor and storage near ISP locations.
6. **Test points don't all look alike** – The belief that all EDL test points were two equally-sized round pads located next to each other and were triggered by ground, is false and is what kept me and others from locating them on many devices. ZTE, Alcatel, Motorola, Samsung, and LG are examples of OEMs that stray from the well-known round pad many times.
7. **Test points are safer than eMMC/UFS faults** – Generally, this is true, but it really has little to do with EDL being created from those locations and more to do with logistics. Test points are more robust, easier to access and connect with the trigger, generally require less teardown of the device, and breaking or damaging a test point normally does not interfere with the functioning of the device. So, yes, safer when compared to shorting an ISP location.
8. **EDL Mode can be created on a running phone** – Not really. Although you can apply a fault or trigger on a running phone and then trigger EDL with the power button, you are just restarting the phone. EDL Mode occurs during the booting of the device and is an alternate boot, which means applying a short to ISP locations or voltage to a test point on a running phone will generally do nothing. Many times, even if the phone is not running but is already connected to USB, shorting or connecting the test point to the trigger will not create EDL Mode. Most EDL procedures involve creating the short or placement of tweezers on a device in the off state and then connecting to USB or applying power.
9. **EDL Mode can be created without connecting to USB** – This is true depending on the device. EDL Mode can be created by use of the power button while the phone is not connected to USB, if the device has a working battery. Triggering EDL by faults, test points, EDL cables, and ADB can be accomplished without USB. The device should be in EDL Mode and detected when connected to a PC. However, sometimes connecting a device already in EDL can knock the device out of EDL.
10. **EDL Mode means an EDL extraction will work** – No. Most modern devices running Qualcomm processors can be placed in EDL Mode. Absent some exceptions, only specific devices with digitally signed Firehose programmers can be extracted via EDL Mode.
11. **EDL Mode is always created intentionally** – No. A device in EDL Mode can mean something is wrong. It can be damaged due to water or other trauma. A device in permanent EDL Mode is not a good thing. I cover that in detail in another section – Bad EDL Mode.
12. **All EDL extractions result in readable user data** – No. Non-decrypting EDL extractions will result in a physical dump of encrypted user data. With few exceptions, that data can't be decrypted once it is extracted from the device. Generally, if the device is extracted in the off state, no decryption occurs. Therefore, decrypting extractions involve booting a device during the extraction process.

3 Testing, locating, probing and pinning for test points

Before I go into more detail on the location of test points on various manufacturers it is important to describe my methodology regarding what I identify as an EDL test point. How do I pronounce something a test point if it doesn't look like a traditional test point? How do I know the point I am using to create EDL Mode is an intentionally created, intentionally placed, test point as opposed to some location responding to a voltage spike that unintentionally or accidentally creates EDL Mode? Why does it matter?

3.1 Test points are intentionally designed to create EDL Mode

This heading may seem like a statement of the obvious, but it differentiates test points from eMMC/UFS faults. Faults occur when Data0-7, CMD, or CLK lines or their exposed locations are shorted by accident or on purpose. EDL Mode can also be created by overloading the device with high voltage (not recommended). In those situations, Qualcomm designed the processor to default to EDL Mode when something is wrong, either to avoid further damage and/or allow a technician to diagnose and fix the problem. I am using a significant portion of this document to discuss damage and other circumstances that can cause and create EDL Mode accidentally. This is because there are a significant number of examiners who experience issues with damaged devices in relation to EDL Mode. It can be frustrating and confusing when it happens. The next few pages might be as interesting as reading the warning label on a lamp. If you don't read it now, it may be beneficial to refer to it in the future when something goes wrong. You may also come to the realization of why an EDL extraction you attempted in the past, failed. It may have failed because you created permanent EDL Mode and not temporary EDL Mode.

3.1.1 Bad EDL Mode

As an examiner, if you receive a device in your lab that is already in EDL Mode when you connect it to USB, there is usually a problem. It means something is shorted due to corrosion or there is some physical damage to the phone. Some phones can be in EDL Mode with no physical damage yet are stuck in EDL Mode even when all power is removed. When being presented with EDL Mode that you didn't request, it would be a mistake to assume something or someone has done you a favor, thus relieving you of the requirement to research or create EDL Mode yourself. If the device is already in EDL Mode, your problems are just beginning. The problem is that the phone will not extract if you don't remove the condition that created EDL Mode. Thus, as I wrote in my EDL Guide, "Don't forget to remove the short before you hit continue." The extraction will fail if you don't. Broken Data, CMD, and CLK lines can create permanent EDL Mode. Chip the storage from some devices running a Qualcomm processor, without damaging the processor. Plug that device into USB after you remove the storage. If the USB port is functional, it will likely be in EDL Mode. The processor can't find the storage you chipped and defaults to EDL Mode. There are a few devices that will extract via non-decrypting EDL methods even if they are permanently shorted. But permanently shorted phones won't boot, so that means no decrypting extraction in the UFED.

3.1.1.1 Accidentally creating permanent EDL Mode

It is possible to damage devices by inadvertently creating permanent EDL Mode while trying to temporarily create EDL Mode. When this happens, it may not be readily apparent to the examiner. The usual way this occurs is when an examiner is attempting to use eMMC or UFS faults to create EDL Mode. It can occur while soldering or while using needles or tweezers. Resistors can be damaged, moved, or bridged and thus create a permanent ground or, in some cases, a permanent voltage bridge. These are some of the reasons I created Cable X, which will be touched upon later.

3.1.1.2 Permanent EDL Mode not noticed by examiners

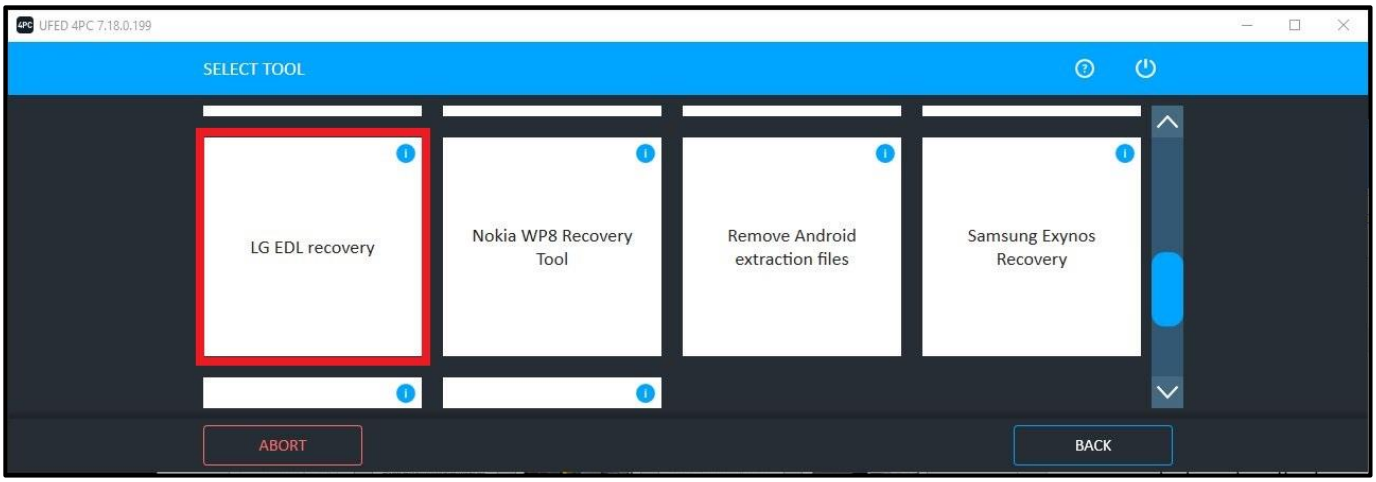
The reason EDL Mode created by damage may not be noticed is that it typically occurs while the examiner is repeatedly attempting an EDL extraction. Thus, seeing EDL Mode after you short a device is what you expect so what you don't see is that your actions are redundant. The device is already shorted by damage and then in EDL Mode when you connect power, but it won't come out of EDL Mode and thus cannot respond to commands given by forensic software. These devices will often fail in the later stages of a UFED EDL extraction – stage 6 of a decrypting extraction via the UFED or stage 2 of a non-decrypting extraction – as opposed to devices not supported for EDL that will normally fail in stage 1 for both decrypting and non-decrypting extractions.

Therefore, an examiner notices the failure and attempts the EDL extraction again by using the needle or tweezers or applying the short with the soldered wire. Of course, the phone is in EDL Mode when the extraction starts and thus reinforces the examiner's belief that their procedure is sound, and the failure must be due to something else. The examiner has created a

situation where the device will always be in EDL Mode when power is applied. EDL Mode will keep a device from booting, so if your device is not booting during an EDL decrypting extraction in the UFED, permanent EDL Mode may be the reason. There are some devices that will dump during a non-decrypting extraction with a short still applied – they are the exception and not the rule.

3.1.1.3 Recognizing and removing permanent EDL Mode – no physical damage

The processor has no memory – generally. In other words, when you short a device and it goes into EDL Mode, the processor is just following protocol. When you disconnect USB and battery power from the device, EDL goes away. When you connect the device again, the processor begins its normal protocol from the beginning and doesn't remember if it was in EDL Mode just a few moments earlier. If there is no physical short of the storage or the D+ USB line and you are not applying voltage to a test point, the processor goes on about its business and boots the phone. Thus, there is no limit to how many times you can place a device in EDL Mode. There are exceptions to this normal behavior with processors in which some devices can get in a permanent EDL loop even though no "physical" shorting of the device occurred and even after all power is removed. The UFED offers a solution to fix this on some LG devices under "Device Tools" in the UFED. See Mastering EDL Mode – LG EDL Recovery Tool - for more details on this procedure. This can also be fixed sometimes just by applying power and button combinations repeatedly until the device comes out of the loop and boots.



3.1.1.4 Recognizing and removing permanent EDL Mode – physical damage

Recognizing permanent EDL Mode caused by physical damage is easy. Just stop trying to create EDL Mode, remove all power and the battery, and then connect the device to USB with the UFED 4PC closed. If the device goes into EDL Mode without the use of the needle, tweezers, or otherwise shorting the device, you have a problem. If this occurs right after your physical tampering with the device to create EDL Mode, it is likely you did something to cause a permanent short. Using a needle or tweezers with too much force can tear through the protective layer or coating of the logic board and exposes the ground layer. If the CMD, CLK, or one of the Data lines or resistors contacts that ground layer you exposed, the device will default to EDL Mode every time power is applied. Dislodging these resistors with too much force can also cause the device to be in EDL Mode each time power is applied. Trying to plug a stubborn micro USB cable into a phone while delicately holding a pin on a tiny CMD resistor with your other hand can be tricky and it doesn't take much force to dislodge a resistor with a needle or tweezers. All these reasons were part of my motivation for creating Cable X. See this sample of how Cable X creates eMMC faults accurately and safely.

3.1.1.5 Verifying and fixing permanent EDL Mode caused by physical damage

Once you have identified the phone is in permanent EDL Mode and that it is likely due to shorting procedures you used, use a multimeter to confirm that diagnosis. For example, let's say you tried to short the CMD resistor with a needle or tweezers and now the device won't come out of EDL Mode. Set your multimeter to check for continuity. Remove or disconnect the battery and disconnect the phone from USB. Place one probe on any part of the phone that is ground and the other on the

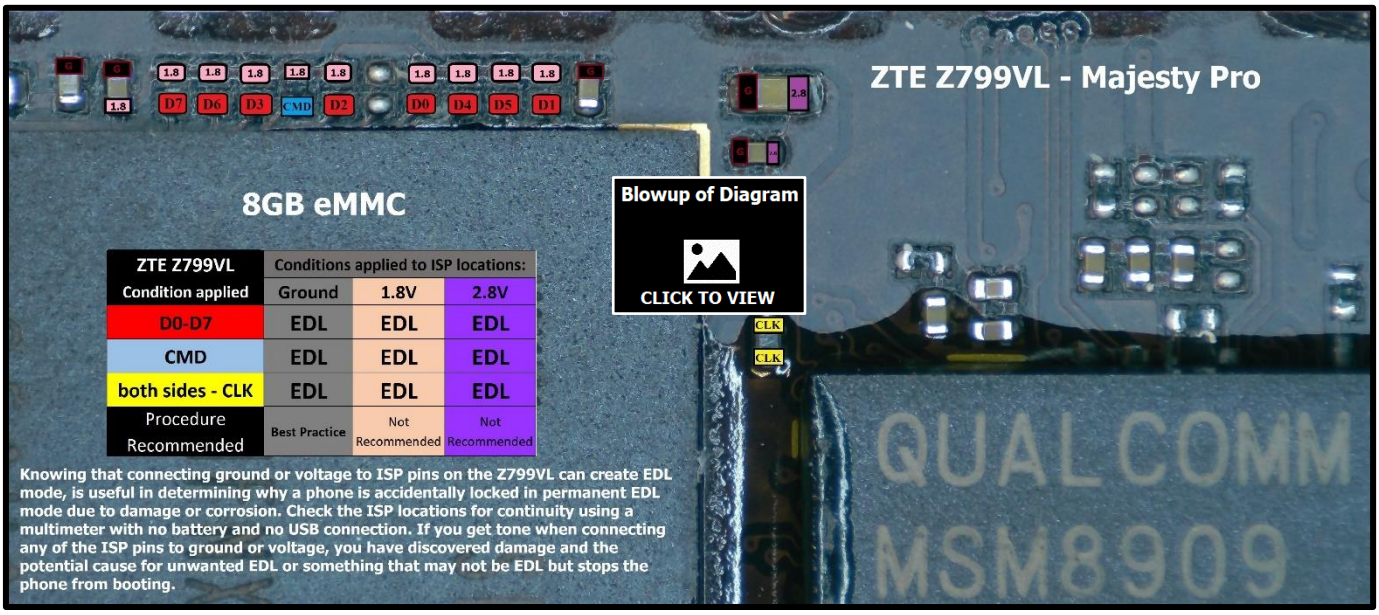
CMD pin. If you get tone, then you have a short that should not be there. You have now likely located the problem. You need to remove that short. How that is accomplished depends on what you were doing when it occurred. It can sometimes be fixed with a soldering iron and flux or very gently moving the resistor. A microscope is very beneficial for this situation. Some phones will boot with the CMD resistor removed. Some phones will boot with the CLK resistor removed and replaced by a solder bridge. Depending on your level of experience and the equipment you possess, it may be necessary to seek some assistance. This is another situation in which skills with ISP, JTAG, and Chip-Off training are very beneficial. You may need to get a test device of the exact model and see if it can function and boot without certain resistors or other modifications.

3.1.1.6 Using ISP pinouts to look for shorts causing unwanted EDL Mode

Yet another reason for ISP, JTAG, and Chip-Off related training is diagnosis of devices that are stuck in EDL Mode or otherwise won't boot. Knowing the ISP pinout of a device can give you an idea where to look first if you have a device stuck in EDL Mode. Remember that all the Data lines D0-D7, CMD, and CLK can all be shorted to create EDL Mode. These are not the only places to look but they are a good start. Shorting voltage locations can also create EDL Mode and even though it is not best practices and downright risky, it can occur inadvertently. This is useful information to know for purposes of trying to fix phones that won't boot or are stuck in EDL Mode.

3.1.1.7 eMMC faults created by applying voltage to ISP points – diagnostic knowledge only

If I were making a TV show this is where I would say "**Please don't try this on evidence**". I am not telling you that applying voltage to ISP points to create eMMC or UFS faults is what you should do. I am telling you that voltage applied to ISP points can create faults, which includes EDL Mode. This knowledge is meant for diagnostics and is not suggested best practices for extracting devices. So, don't do it unless you have a test phone you don't care about. You can do whatever you want with a test phone. The recommended and proven method for creating eMMC and UFS faults for EDL Mode is applying ground to CMD, Data, and CLK. Voltage will work on many phones, but ground is recommended by every expert and is also what I use. Voltage applied to ISP locations can damage some phones permanently. Ground locations are safe and abundant.



When you are dealing with a piece of evidence that should not be in EDL Mode or one that you damaged and won't come out of EDL Mode, you need to know all the possibilities. If you were shorting the CMD pin to ground to create EDL Mode and now your device is always in EDL Mode, you may have created a permanent ground to CMD or you may have created a voltage bridge. Either of those things can create EDL Mode. The other side of the CMD resistor is most often VCCQ (1.8 volts). Bridges can also create other issues much worse than EDL Mode which can prevent devices from booting or permanently damage devices. I tested a Z799VL (test device) just to see what would create EDL Mode. Grounding every ISP point created EDL Mode, that includes all 8 Data locations, CMD, and both sides of the CLK pin. I then applied voltage (VCC and VCCQ) to those same locations and that also created EDL Mode repeatedly. There were no side affects to any of this and the phone still

functioned afterwards, but that is not meant to be an endorsement of doing this or a suggestion that it is safe. Apply voltage to ISP locations on some other phones and you will break them. When it comes to evidence, don't do it. When it comes to evidence that is permanently in EDL Mode, understand that ground and voltage bridges can both create unwanted EDL Mode.

3.1.2 Test Points are safer than faults

Because test points are intentionally created and specifically placed for creating EDL Mode by the manufacturer, we can assume they are somewhat safe. Creating EDL Mode via eMMC faults is safe in that there is no limit to how many times you can short a device into EDL Mode if you don't damage the phone, physically, when you do it. When phones get damaged while attempting to create EDL Mode, it is not EDL Mode that damages the phones. We damage those phones with our clumsy or hasty methods of creating EDL Mode.

3.1.3 Test points are generally larger and more conveniently located than fault locations

If you use too much force when soldering or using tweezers or a needle to connect a resistor to ground, you can damage the phone, create a solder bridge, or otherwise make an extraction impossible until you fix what you broke. Test points are generally designed to be larger, more accessible, and more robust than ISP locations. They are more "cop proof" than ISP locations. I am sure most cops understand what I mean by that.

Test points are often located face-up when exposing the logic board. Test points are often outside of the cover of heat shields. The trigger is most often conveniently located next to the test point and at the same elevation as opposed to using a heat shield to ground a CLK resistor with tweezers. The calculation has been made by the OEM regarding how much voltage will be applied to the test point to create EDL Mode. The test point is presumably placed there so it is not necessary for a technician to create a fault by shorting. Of course, I am taking some liberties with assumptions because manufactures don't provide us with a guide to accessing their devices or tell us exactly why they placed pads where they do. However, you can compare test points to JTAG taps - specifically placed and designed for access and designed to respond to specific input.

3.2 Hunting for test points

Sometime in January 2019, I set out to do some research on a variety of devices for the purpose of locating test points. I started with specific devices and brands not known to have test points. Some phones go into EDL Mode via button combinations, ADB commands, or EDL cables. All those methods can be accomplished without device disassembly. Device manufacturers can shut off those methods of creating EDL Mode. When those methods don't work, some devices have test points. Test points require disassembly but are relatively easy to use and access in most cases.

When everything else fails and test points are not found, storage faults are all that remain. For some phones, this means removing and flipping the logic board and possibly soldering. For encrypted devices that must boot, the battery must be in place after attaching a short wire to an ISP location. It can be difficult for those not used to the procedure and some devices are just plain tricky. I asked myself if it was possible there were test points located on devices that myself and others have disassembled many times and we were just not seeing them? I decided to fall back on a technique and skill that has helped me locate ISP locations, JTAG TAPS, alternate USB access, home button, power and volume locations. I am talking about chipping and pinning.

3.2.1 Pinning for test points involves skills from the past and present

Before getting into how I pin for test points I must take the opportunity to emphasize the need for training and practicing ISP, JTAG, and Chip-Off. There is not a day that goes by in which I don't rely on that training and practice when doing forensic exams. The equipment and skills needed for ISP, JTAG, and Chip-Off will allow examiners to diagnose and repair damaged devices, take full advantage of EDL extractions, and oh yeah, also perform ISP, JTAG, and Chip-Off, all of which are still very relevant skills and solutions I use on many devices today. Absolutely the most beneficial class I have ever taken was a JTAG class taught by Kim Thomson in 2014. Chris Weber later introduced me to ISP and Chip-Off. I learned more from those two individuals than anyone else in forensics.

3.2.2 Going from the known to the unknown

Pinning for ISP locations on a device involves chipping the storage on a phone and then starting from known locations under the storage and using a multimeter to locate positions on the device that can be accessed without removing the storage.

Those locations are mapped and that allows other examiners to extract data from the storage on the same model of phone without having to chip-off their evidence. Of course, to pinout a phone for ISP locations, someone had to identify the known locations on eMMC chips first. Using that same concept, I decided to see if I could locate test points with the same techniques.

3.2.3 Everything is under the processor

The processor is the brain of the phone, so everything eventually leads to some point under the processor. Thus, if you have an ISP pinout of a processor, you can chip-off the processor to extract and/or create an ISP pinout of a phone. The reason we are able locate ISP points is because the lines appear on the surface of the logic board on their way from the storage to the processor. I have pinned many processors for JTAG locations and that allows me to chip a processor on a device and trace those points to identify the JTAG TAPS on a phone. When the user of a device pushes the home button, volume button, or power button on a phone, there is a line that leads back under the processor. The USB data lines lead to points under the processor. Knowing all these locations has helped me get into many devices and allowed me to mimic commands to devices that have been torn in half. Test points that create EDL Mode are no different. If connecting one point to another point creates EDL Mode, that means a signal is sent to a specific pin under the Qualcomm processor. Locating that pin on a processor using a phone with known test points, can lead to hidden test points on other devices running that same processor.

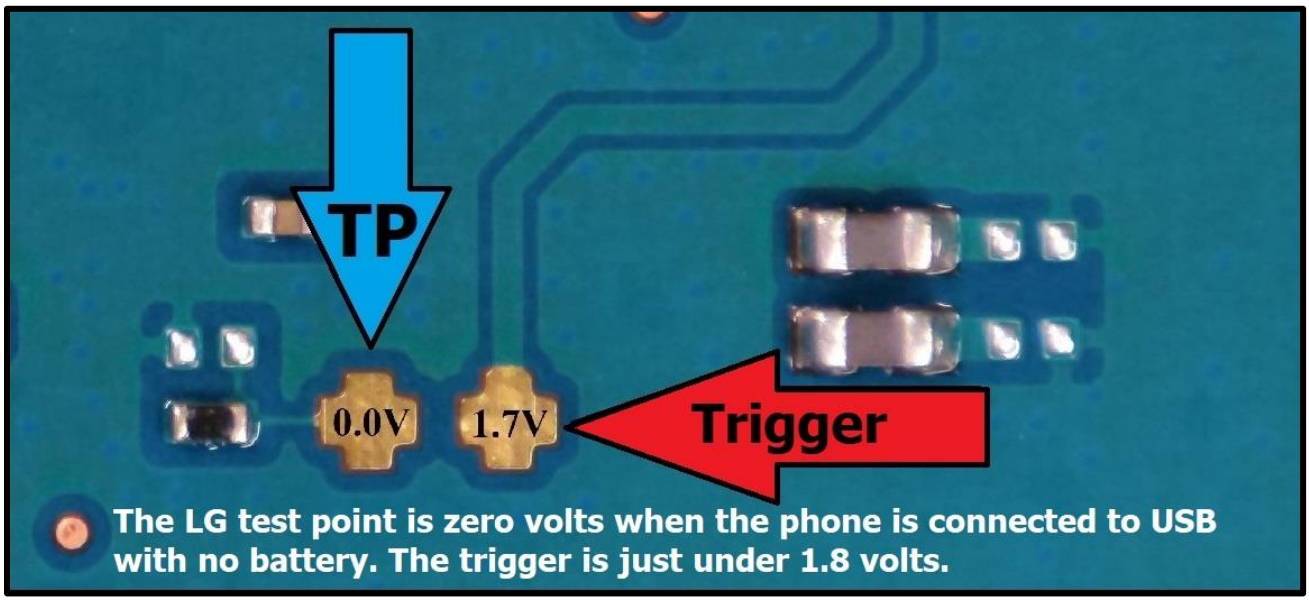
3.2.4 Using ISP and JTAG pinning skills to locate test points

When I learned how to pin phones for ISP, someone had already located the ISP points on eMMC storage so that I knew where to start with my multimeter. When I started looking for JTAG pins under processors, I had to locate those for myself. I did that by using phones with identified and proven JTAG taps, chipped the processor, and used the multimeter to go from the TAP to the pin under the processor. To pin the Qualcomm APQ8064 for ISP and JTAG locations, I chipped the storage and processor from the Samsung SGH-i537. I used the known pinout of a BGA153 storage to locate the ISP points on the processor and I used a known JTAG TAP diagram of the SGH-i537 to locate the JTAG pins under the processor to create the diagram of the APQ8064. Thus, using the pinout of the Qualcomm APQ8064 pinned from a Samsung SGH-i537, I can locate ISP or JTAG taps on any other phone running that same processor. All I must do is chip the processor on an exemplar. I used this same strategy for test points.

3.2.5 LG devices were the starting point for pinning processors for test points

I started with well-known test points on phones and identified attributes of the test points and the trigger. The most reliable way to check for voltage on most devices is with the USB connected with no battery. There are some exceptions to that rule, as with Samsung devices, discussed later. When testing for ground, disconnect the battery and unplug from USB and check for continuity between the unknown pad and grounded portions of the phone. Ground pads can most often be spotted with a visual inspection as you can see that the pad is connected to the ground layer just under the outer coating of the logic board's surface.

LG test points are the most recognizable of all the manufacturers. On most modern LG phones, the test point is always next to the trigger and they are both in the shape of crosses. I identified what each test point did. I quickly figured out that one of the crosses was connected to many locations by continuity and those locations were voltage. From basic ISP skills, you know



that there is typically only one CMD, one CLK, and one Data0, but there are multiple voltage locations. A "test point", something that tells the processor to default to EDL Mode, should be unique also. Therefore, the "test point" is the other cross that is not connected to everything else on the phone. That test point should lead to one point under the processor and thus marking that point on the LG phone should help me locate the test point on other phones running that same processor. The next step was to chip-off the processor on the LG phone and with a pin I identified as the "test point" and use my multimeter to go back to the processor.

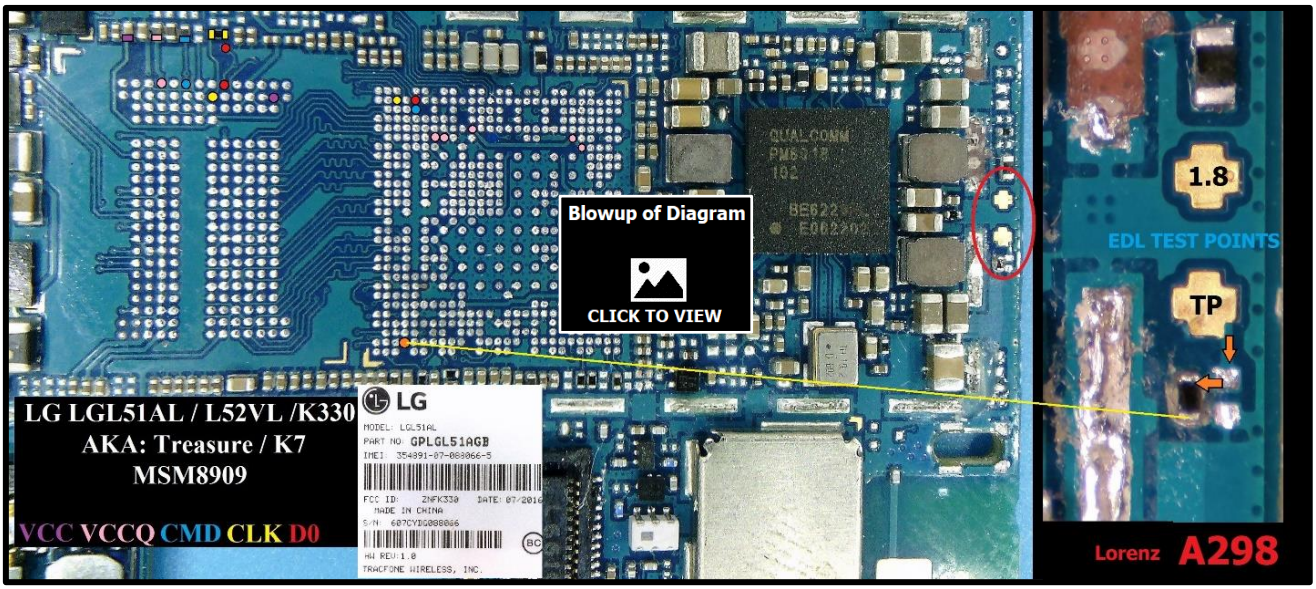
3.3 Reverse pinning the LG51AL to locate the test point on the MSM8909

Understanding how the test points work on LG phones lets me know that the actual test point will test for 0 volts and the trigger will test for 1.8 volts. That means I can identify which cross is the test point on any given LG phone by removing the battery, connecting to USB, and testing for voltage with a multimeter. Then I know which pin to follow to the processor to identify the pin under that processor that is the test point.

When pinning for locations under processors, it is important to remember that you may not get continuity directly from the test point from which you are tracing directly to the final destination under the processor. The same is true in reverse if you are looking for a test point from a known location under a processor. With the LGL51AL, there is a resistor in the path to the processor that will not allow your meter to sound when pinning. During the pinning process, when you come to the resister, pin from the opposite side from where you began and then continue identifying the path until you reach the processor. On a working phone, you place tweezers on the test point and the voltage location marked and then connect the phone to USB. The voltage travels through your tweezers, into the test point pad, through the resistor, and finally to the processor. The processor has been programmed to interpret this signal as a need for EDL, and defaults to EDL Mode.

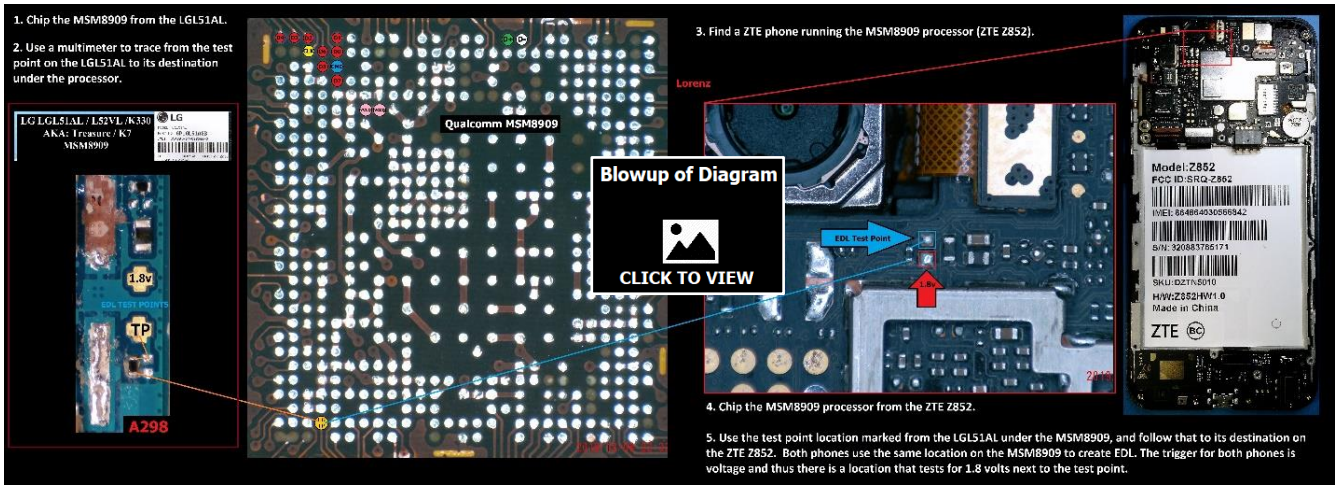
3.3.1 Using the LG test point under the MSM8909 to find test points on other brands

There is no need to go looking for test points on other LG phones running the MSM8909 with the marked location under the processor. LG makes it easy to identify the test point on most of their more modern phones. They are all crosses. The point of identifying the LG test point on the MSM8909 was to use that information to locate test points on other OEMs not known to have them. Chipping off the MSM8909 processor from a ZTE phone should allow me to trace my marked test point pin from the LG phone running the MSM8909. The idea was to test if that pin would lead to an unknown test point on the ZTE phone. This is the same concept used to pin a phone for ISP locations – going from the known to the unknown by chipping the storage on a phone. Of course, by chipping the processor on a test ZTE phone to find a hidden test point with a multimeter would mean that I would need a second test phone to see if that location created EDL Mode. A phone cannot go into EDL Mode after removing the processor.



3.3.2 Using the known LG test point to pin for the test point on the ZTE Z852

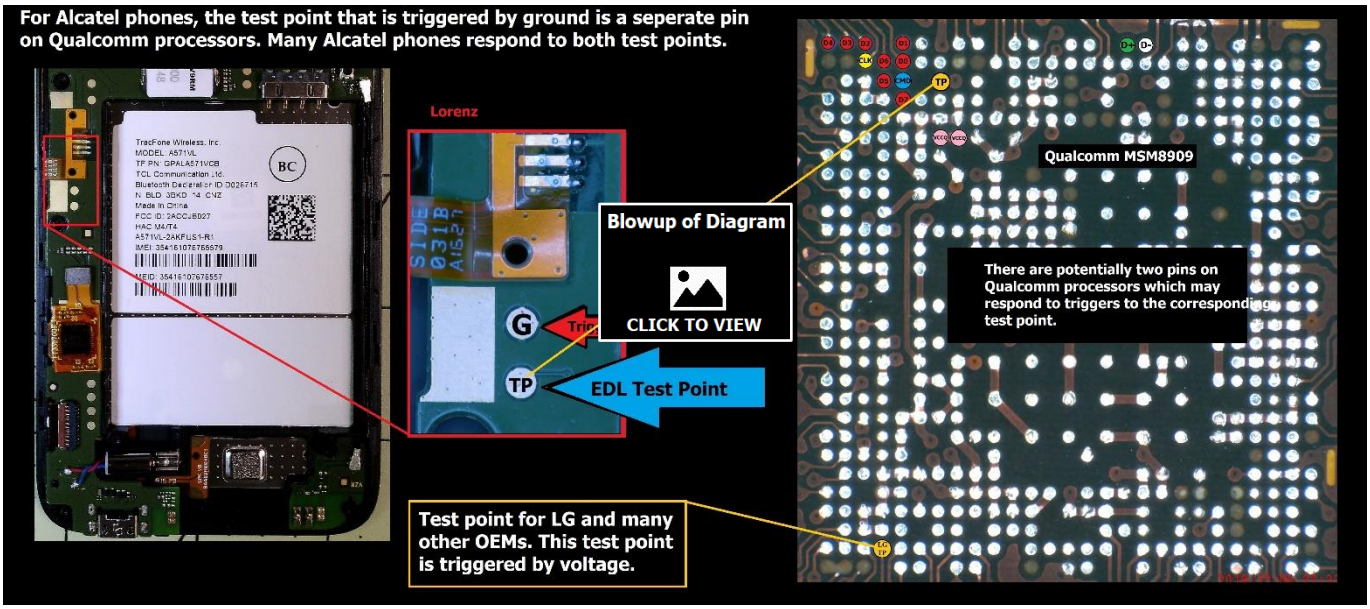
By “reverse pinning”, I mean going from the known test point on an LG phone to the corresponding unknown location on the MSM8909, and then to the unknown test point on another phone. I selected a ZTE phone running the MSM8909 processor. The ZTE Z852 is running an MSM8909 processor. The UFED will extract this device, however, the only way previously known to get this phone into EDL Mode was eMMC fault injection – shorting one of the ISP locations. That required flipping the logic



board as the processor and eMMC are located on the underside of the board. There are two other variants of this phone with almost identical logic boards (ZTE N9137, Z557BL). Button combinations may or may not work on any of these models. If a test point could be located, it could improve the extraction process by eliminating the need to short ISP locations and possibly eliminate the need to flip the logic board. When I reverse pinned the Z852 from the processor to the test point location I realized it was right under my nose the entire time and in many photographs, I had taken in the past. Don't get me wrong, the points are small, but once you know what you are looking for, they are somewhat distinct. I realized that incorrect assumptions I had made regarding where test points should be located, what they should look like, and on what devices they would and would not be, kept me from finding them before.

3.3.3 Alcatel devices with known test points help to identify a second test point location on the MSM8909

If Alcatel phones are the exception to the rule regarding test points, is there a second location on Qualcomm processors specifically for that exception? One test point on Alcatel phones behaves the same as every other OEM – the test point is no voltage and the trigger is generally 1.8 volts. The most commonly known and used test point on Alcatel phones is the one I refer to as the exception to the rule – the test point is 1.8 volts and the trigger is ground. I sought to identify that test point using the same technique as I used to identify the first test point. For that technique, I needed an Alcatel phone running an MSM8909 processor, with a known test point triggered by ground. There is a second test point for Alcatel phones.



3.3.4 Is the processor built for the phone, or the phone built for the processor?

The argument can be made either way to some degree but more likely the OEMs base their design on what processor they plan to mount. If nothing else, the language used in the September 15, 2015, press release from Qualcomm makes this clear. The announcement of the Snapdragon 410 and 210 shipments, which include the MSM8909 and MSM8916, helps identify the size and scope of the patterns forensic examiners are familiar with regarding these two processors. Note that Qualcomm states new devices are based on the processor to be used.

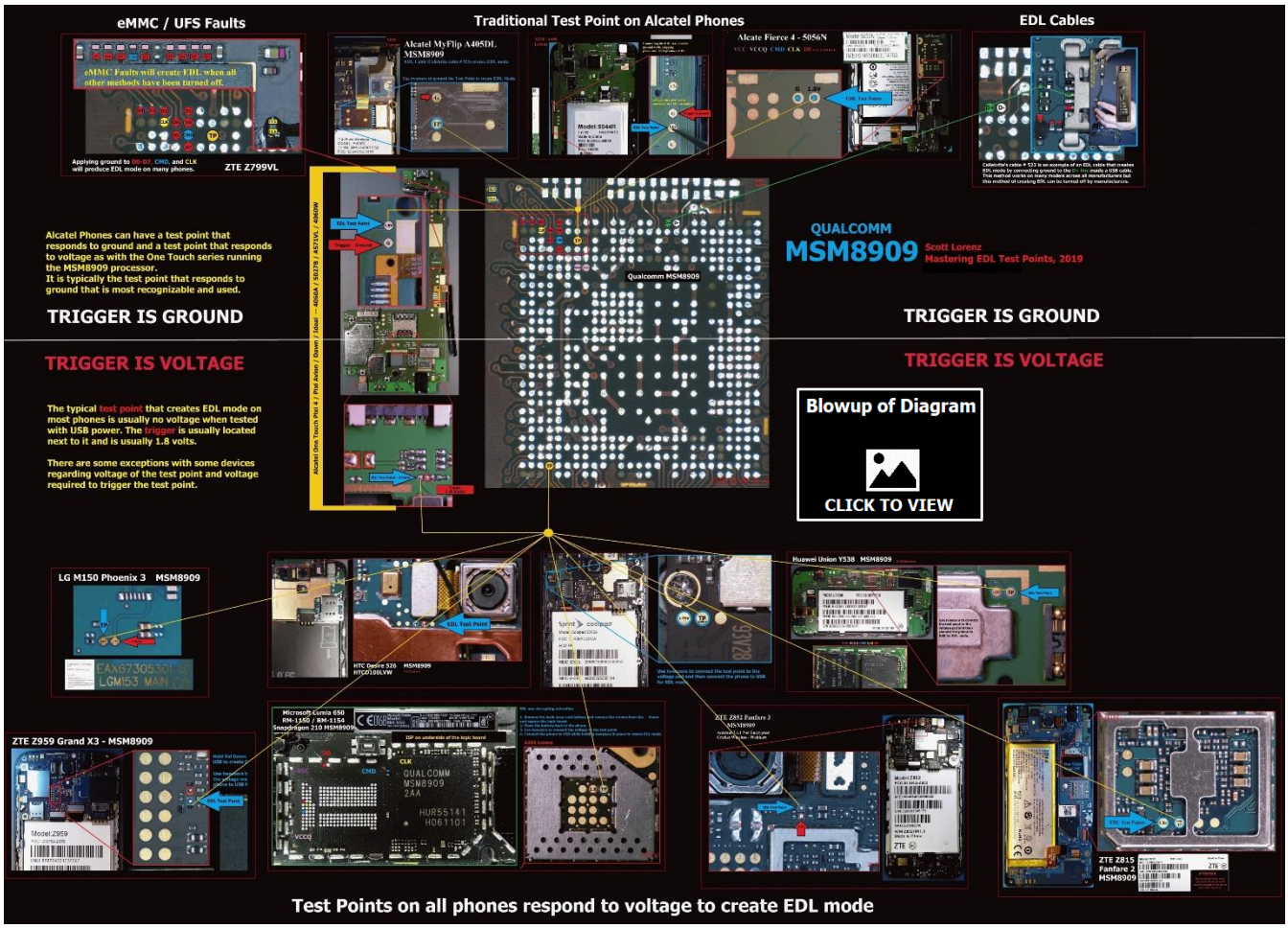
> *"The Snapdragon 410 processor has also shipped more than 200 million units globally from more than 60 OEMs. Additionally, the Snapdragon 210 processor has been included in more than 200 designs either shipped or in the device pipeline….**New devices based on Snapdragon 210 processor** include the Alcatel OneTouch Pixi 3(5), ZTE A460, and Huawei Bee 4G and Honor 4A. New devices based on the Snapdragon 410 processor include Motorola Moto G, Asus Zenphone 2 (ZE550KL), HTC Desire 510, HTC Desire 626, Samsung Galaxy Mega 2, Xiaomi Redmi 2 and BLU Life One. We anticipate more devices in the coming weeks and months."* (https://www.qualcomm.com/news/releases/2015/09/15/qualcomm-announces-new-milestone-snapdragon-410-and-210-shipments)

Nature and man take the path of least resistance and that creates patterns by necessity, which can be recognized and used. The USB D+ and D- pins are always in the same location on the MSM8909 processor regardless of what phone in which it is mounted. OEMs design phones to fit the MSM8909 if that is the processor they plan to mount on their phone. Thus, the OEM must design the USB lines to run from the USB port to the precise locations where the D+ and D- pads will be to align with the MSM8909's BGA. Qualcomm provides OEMs the complete datasheets for their processors to coordinate this effort, otherwise chaos would ensue. Some processor datasheets can be found on the internet while others are hard to locate or are highly guarded.

The practice of the OEM designing their device to match the BGA of the planned processor is the same for all the ISP pins. D0-D7, CMD, and CLK are always in the same location on the MSM8909. The MSM8909's BGA is configured the same for LG, ZTE, Coolpad, Alcatel, or any particular brand of phone in which it is mounted. I know the ISP pins will always be in the upper left corner of the MSM8909 and the USB lines are near the top, center. Thus, test points should be the same; that is, follow the same consistency.

3.3.5 OEMs follow the lead of Qualcomm in what pin creates EDL

Why change something that already exists if it works? The EDL pin location on the MSM8909's BGA which is mounted on an LG phone should be the same on a ZTE phone, or a Coolpad, or an Alcatel. If that is true, the rules regarding what creates EDL when it comes to test points should be the same. If applying voltage to the test point on the LG phone creates EDL Mode, that should be true for any OEM using that same EDL pin for the MSM8909 BGA. The same is true for the pin on the MSM8909, used by Alcatel that is triggered by ground. Therefore, one test point location on the MSM8909 is triggered by

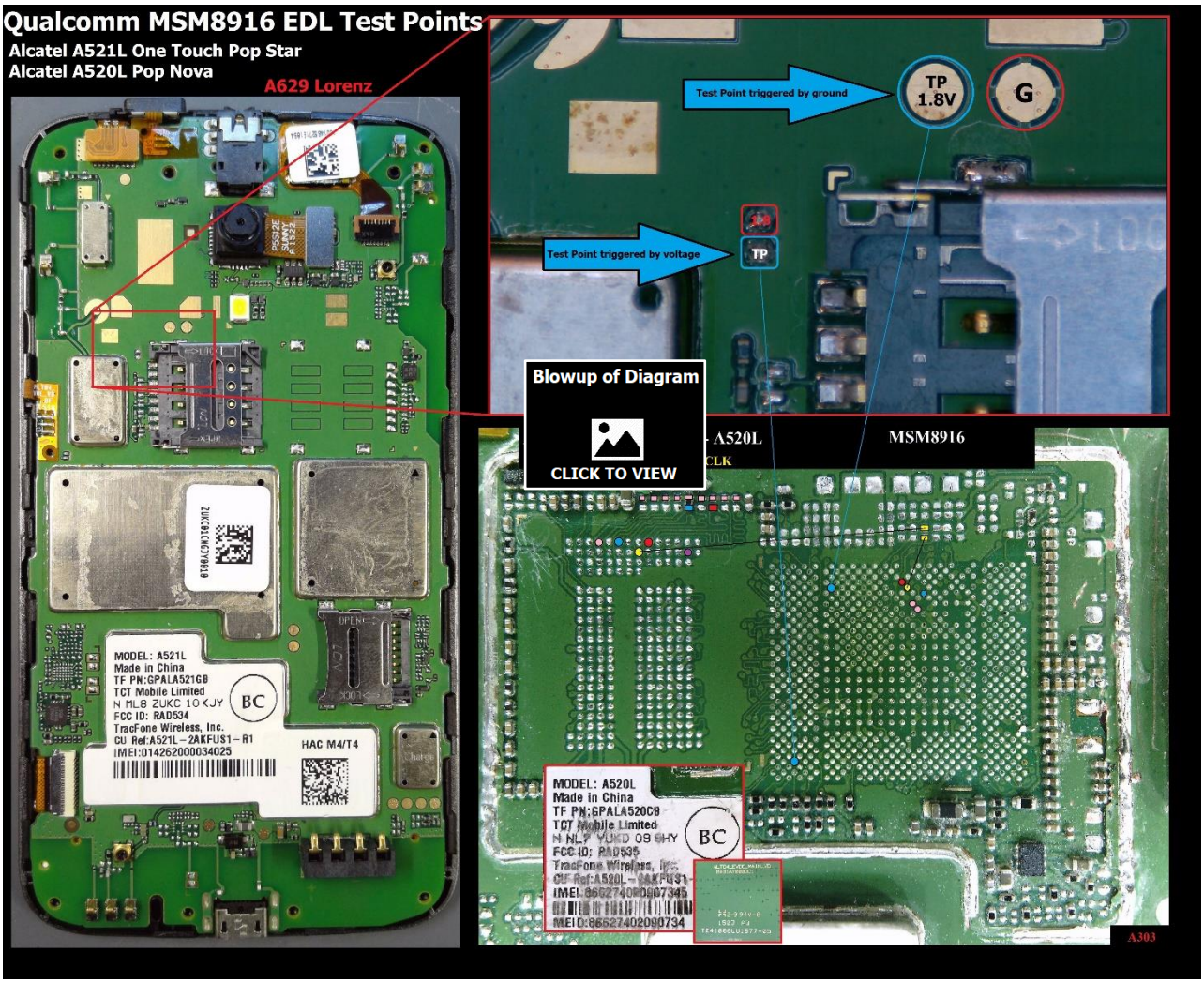


ground and the other is triggered by voltage. It is just like shorting the CMD always creates EDL Mode on any phone running an MSM8909, regardless of what OEM made that phone. Perhaps the best way to appreciate the patterns that exist

regarding processor design as they relate to locations of interests on phones is to see the similarity in different OEMs running the same processor.

3.4 Pinning the Qualcomm MSM8916 for EDL test points

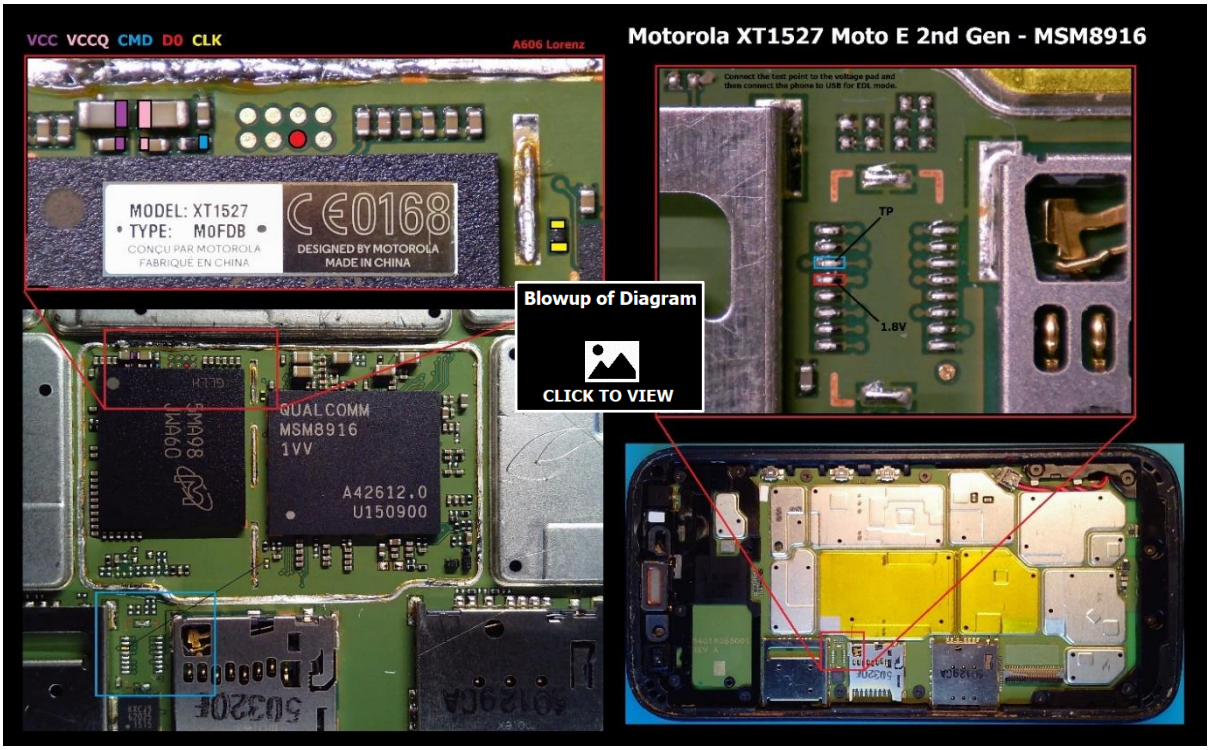
The MSM8909 is the center of the universe when it comes to the number of phones containing that processor and for still being popular in new phones today. Devices running the MSM8909 can be encrypted or not, depending on the device. The MSM8916 is another processor found on many phones, but that processor is not being installed on new phones today. Phones running the MSM8916 are rarely encrypted so that opens options for low-level, non-decrypting EDL dumps. No booting is required, and devices do not have to function to dump. I used the same process for identifying EDL pins on the MSM8916's BGA as I did for the MSM8909. I started with devices running the MSM8916 with known and proven test points, chipped the processor, and pinned from the test point to the processor. Test Points on the Alcatel A521/A520 lead to both pins on the MSM8916 that trigger EDL Mode. The test point triggered by voltage on the Alcatel A521 leads to the pin that triggers EDL Mode on other OEMs running the MSM8916, including but not limited to LG, Motorola, Samsung, and ZTE.



3.4.1 Discovering unknown EDL test points on Motorola phones using locations from the Alcatel A521L

After pinning the MSM8916 for the two test point locations, I tested numerous phones to confirm the existence of the same patterns and consistency found on the MSM8909 regarding EDL test points. I also discovered test points where I didn't expect to find them. The Motorola XT1527 Moto E 2nd Generation runs the MSM8916. This device is supported in the UFED for Physical Lock Bypass using a bootloader working on security patches up to May 2017. Phones running the MSM8916 are never encrypted by default so ISP and chip-off is an option. Motorola phones of this specific time are notorious for having very robust heat shields covering just about the entire surface of the logic board. Without practice, it is easy to cause damage to the board when trying to remove these heat shields. I obtained a test device and removed the processor and storage to pin the device for ISP and for the possible location of EDL test points. I used the test points I marked on the MSM8916 when I reverse pinned the Alcatel A521L. I was surprised to find the EDL test point was one of the pads on the unoccupied Molex mounting location. On older phones with processors supported for JTAG, these locations are frequently where JTAG TAPS are located. I noted that the trigger (1.8 volts) was located right next to the test point.

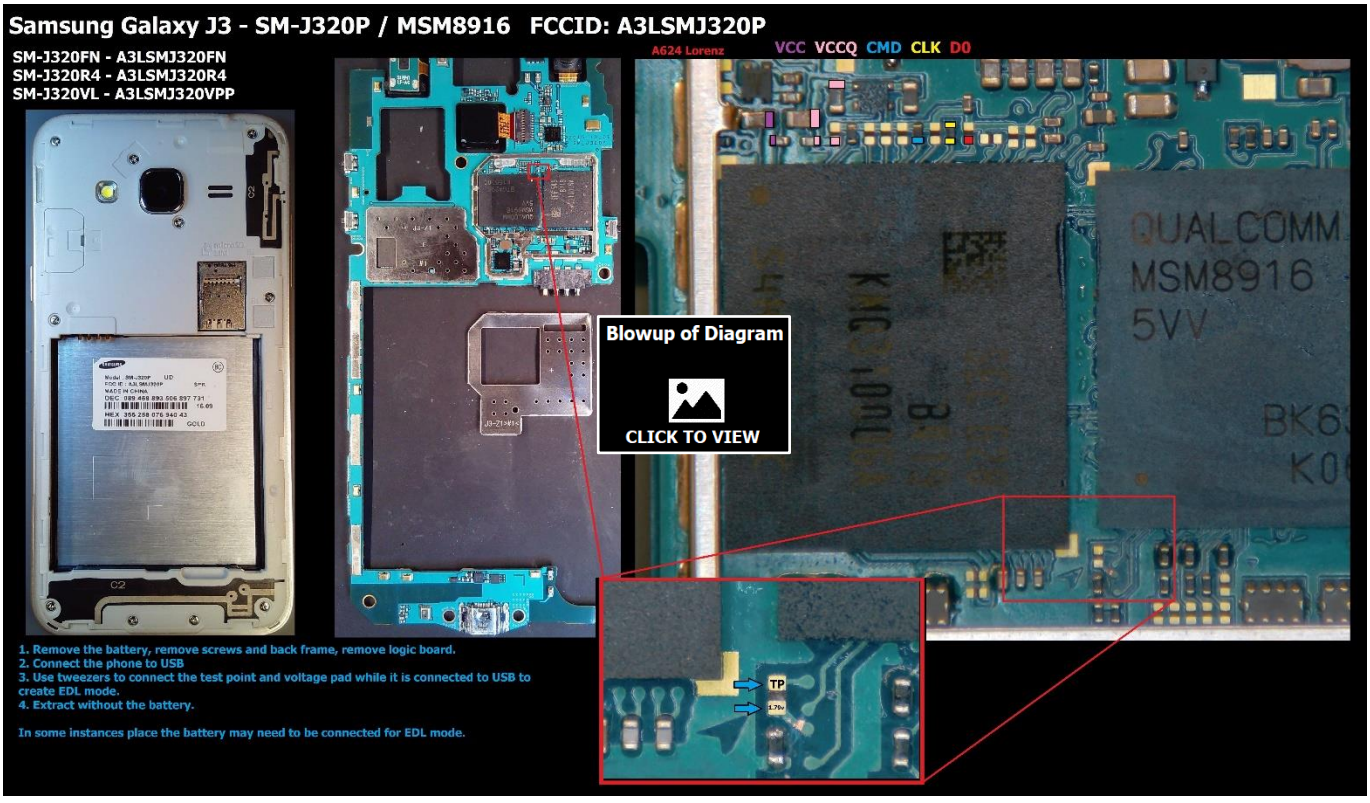
I tested this test point on another working XT1527. These locations are tiny, so I used a SIM tray removal tool to touch the test point and the trigger at the same time while connecting to USB. EDL was created and I extracted the phone using generic non-decrypting EDL options in the UFED. Note from the diagram that the Molex location with the EDL test point is located outside of all the heat shields, thereby putting the examiner in a more favorable position of being able to get a full physical from the phone without the pain of removing armored heat shields and performing ISP. Using the EDL test point in this situation also means the phone is not required to boot, as with the bootloader method, and thus damaged devices can be easily extracted while in the off state with EDL non-decrypting in the UFED.



3.4.2 Using the MSM8916 to pin for EDL test points on Samsung phones

Samsung phones were generally known to not have EDL test points. Many Samsung phones run Exynos processors but a significant number of them run Qualcomm processors to include several models that run the MSM8916. Using the same process, I chipped the processor and hunted for the test point. Once again, I found the patterns and consistency on the MSM8909 were present with the MSM8916. OEMs like Samsung were more likely than not to have a standard test point connected to the same pin on the BGA of the MSM8916 as Motorola and other devices. Aside from some powering and battery issues that differed from other phones, the process was the same. The test points were much easier to access than ISP locations used to create eMMC faults.

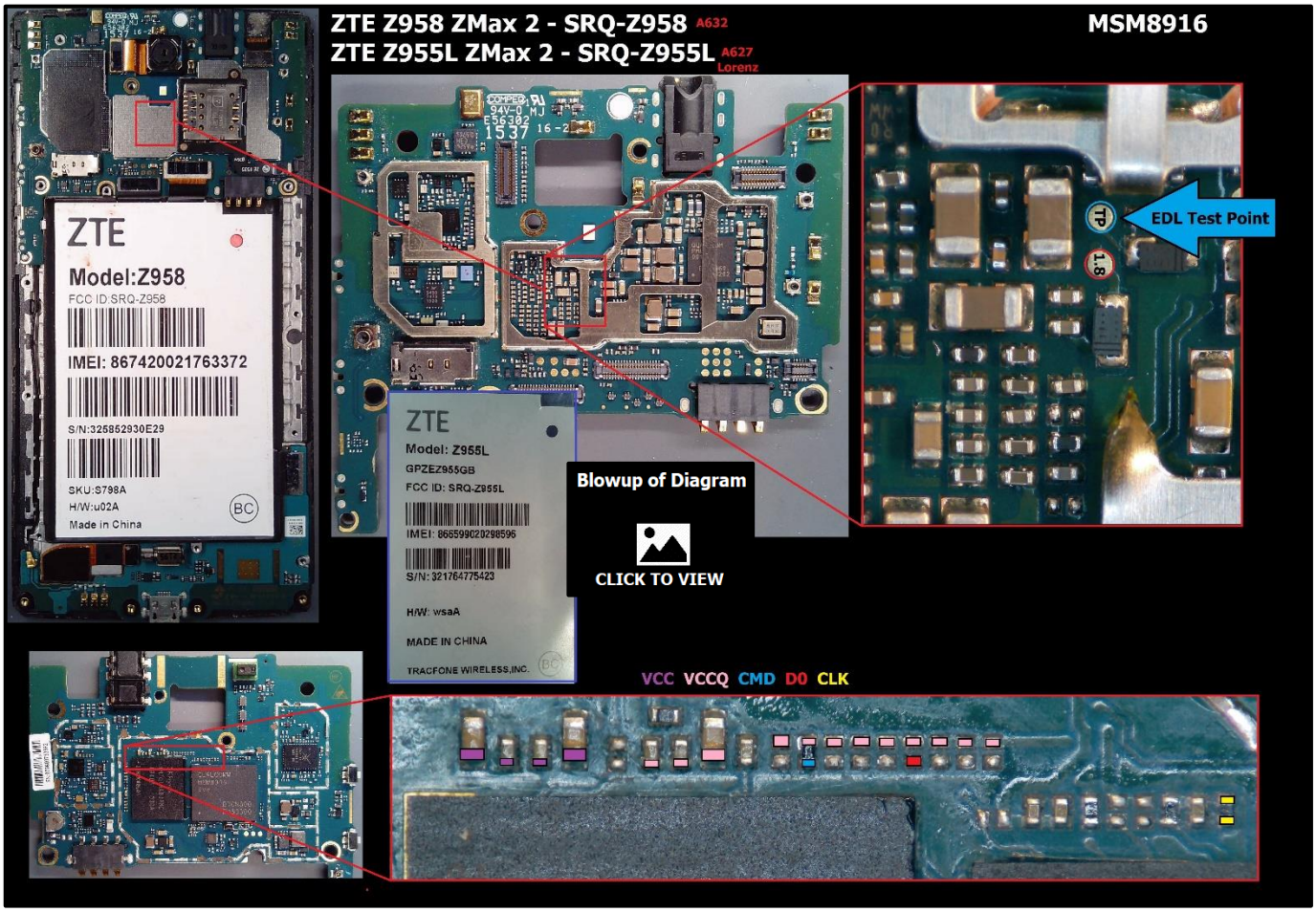
The Samsung Galaxy J3 SM-J320P is supported in the UFED using a legacy bootloader and is supported for EDL ADB if the device is unlocked. However, the generic options in the UFED are powerful tools if you know how to use them and what to look for. If the examiner can find a way to place a device in EDL Mode, a non-decrypting EDL extraction can be easy to perform under generic EDL options. Again, this also allows damaged, non-functioning devices to be extracted without booting. Test points mean no soldering is required and they are much easier to access than ISP locations.



3.4.3 Using the MSM8916 to pin for EDL test points on ZTE phones

Like Samsung phones, ZTE phones were not typically known for having test points. Using the pins from the MSM8916 allowed me to search for not so obvious locations for EDL test points. But “not so obvious” doesn’t mean inaccessible. With most phones, I had originally presumed not to have test points; my research found they were right under my nose all along. I just didn’t look for them.

The ZTE Z958 / Z955L ZMax 2 is supported in the UFED. The Z958 is supported for Physical Lock Bypass using a button combination. The Z955L is listed as supported for physical via ADB. The idea behind EDL test points is to provide options for physical lock bypass when there is no other way, or to provide alternatives when phones are damaged, won’t boot, or generally have issues that make other bootloader methods fail. The ZMax 2 is running the MSM8916 and is not encrypted so ISP is an option. However, the device can be easily extracted using UFED non-decrypting EDL and the EDL test point that is conveniently located on the face-up side of the logic board when the back cover and frame are removed. Test points on ZTE phones follow the rule that the trigger is 1.8 volts.

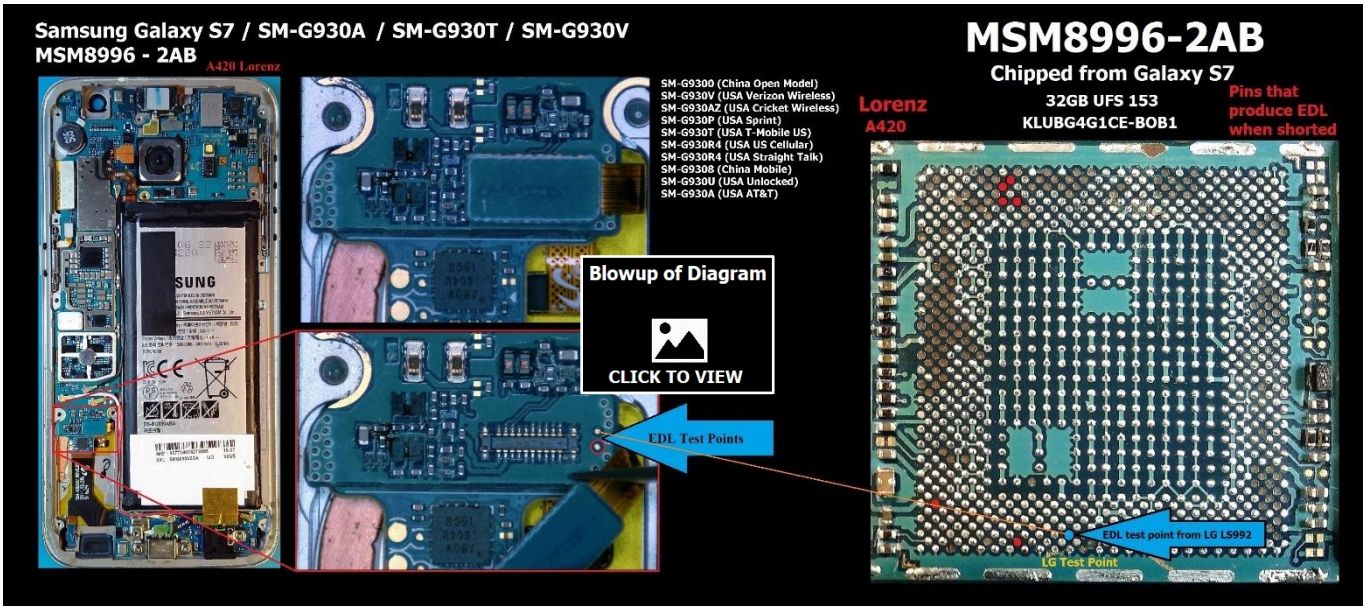


3.5 Reverse pinning the MSM8996 to locate EDL test points on the S7, S8, and S9 phones

Some of you might be thinking that the Galaxy S8 and S9 don't run the MSM8996 processor and you are correct. However, finding EDL test points by pinning one processor allowed me to find locations on other models from the same OEM running other Qualcomm processors. Once I found the test points on the S7 using the pins from the MSM8996, I knew right where to look on the S8 and S9, which run the MSM8998. It also made me feel a little foolish for not finding them before. I had been pinning these devices for UFS fault locations to create EDL Mode. The S7 models enjoy some EDL support via ADB but thus far are not supported for lock-bypass decrypting EDL extractions in the UFED. I mark and pin many phones for methods to create EDL Mode even if there are no firehose loaders for them at the time. There may be a need for EDL Mode on these devices in the future and I have never regretted the use of time I have spent looking at phone through a microscope as I frequently make use of something, I discovered months or years ago, useful to me today.

Most phones running the MSM8996, including all the Galaxy phones use UFS (Universal Flash Storage) instead of eMMC (Embedded Multimedia Card) storage. UFS is faster than eMMC and therefore improves performance on high-end devices. Unlike eMMC storage, UFS is somewhat of a mystery when it comes to ISP. There is generally no publicly available tool or method to ISP devices with UFS storage and no real desire to explore that exploit as devices running UFS storage are all encrypted by default with very few exceptions. Encryption kills ISP and thus no one really explored using ISP for forensic recovery of user data.

With all of that said, UFS is subject to fault injection just like eMMC storage. Devices running UFS storage can be shorted into EDL Mode by shorting one of the data lines, CLK, or CMD. I have pinned many devices running UFS storage for locations that create EDL Mode. I just don't know precisely what I am shorting other than it is what we know as Data, CLK, or CMD. Of course, phones with UFS storage running Qualcomm processors have test points to create EDL Mode. I just didn't know there were any test points on the Galaxy S7, S8, or S9 until I started working on this project - specifically until I chipped the LG LS992 G5 running UFS and the MSM8996. LG phones having test points shaped like crosses makes them a good phone to pin EDL locations on Qualcomm processors and thus be able to reverse pin phones like the Galaxy S7 in search of an EDL test point. Once I located the test point on the S7, it was easy to find it in the same location on the Galaxy S8 and S9.



3.6 General concerns and considerations for probing for test points and attempting extractions

It is possible to locate test points on devices where no previous diagram exists and without chipping the processor on a test phone. It is almost like locating IPS points or JTAG TAPS on devices. Looking for test points involves knowing what is typical for a particular manufacturer and then being able to use a multimeter and process of elimination. Being able to see examples of test points from diagrams of many different devices is helpful. There are rules and patterns that are consistent across manufacturers. The best way to cover this is to talk about what is consistent across all manufacturers and then cover examples of each manufacturer.

3.6.1 The risk of probing for test points

When providing examiners the location of test points I am sometimes asked if it is safe to use test points. I always say yes but there is no procedure that is completely void of risk. Using a test point is safe. Probing for a test is generally safe but it is not without risks. However, I would rather present all of the facts and what I know and allow each individual examiner to make their own determination about whether it is safe to perform a particular action. The answer varies.

Test points, as I have defined and identified in this paper, were put there intentionally for creating EDL Mode. Like any other procedure, care must be taken by the examiner to make sure the proper procedures and precautions are used. ISP and JTAG are safe when properly done but things can go wrong. Disassembly of devices can cause damage if it is done in haste and without prior research. There are many issues surrounding the use of test points in which things can go wrong. As with any other forensic procedure, things can go wrong even if you never take the device apart. The individual examiner must evaluate all the information, options, and risks then weigh that against their own skillset, training, and tools available. The final consideration is how important or critical the evidence on a device is and will the success or failure of an extraction make or break the case?

3.6.2 Get a test phone

This paragraph was not written specifically for test points. I have always been surprised at the reluctance of criminal justice professionals and administrators when it comes to spending a few dollars to obtain a test device. I am trying not to mention any specific profession or title, as I don't think this is an issue endemic to one actor in the criminal justice system. After listening to a five-minute speech about how critical a piece of evidence is to a criminal case, the person giving me that speech then becomes reluctant to a suggestion of purchasing a test device. Suddenly the critical piece of evidence is not worth an expenditure of 50 dollars. There seems to be no reluctance to buy multiple boxes of ammo for practice. The same should be true with forensics when it comes to test devices. I feel better about what I am about to do to a piece of critical evidence if I have already performed that same procedure on another device. If you are about to try something for the first time, or if what you are about to do is "a first" for a device, make sure the device is not the only piece of evidence in a homicide that occurred last night.

3.7 Tips, tricks, and tools

There are some shortcuts, tricks, and tools I have learned and developed by necessity when testing so many devices. There is more than one way to accomplish many of the techniques used in forensics. It is up to each individual examiner to select methods and tools with which they are comfortable or based on the constraints of their departmental or agency guidelines.

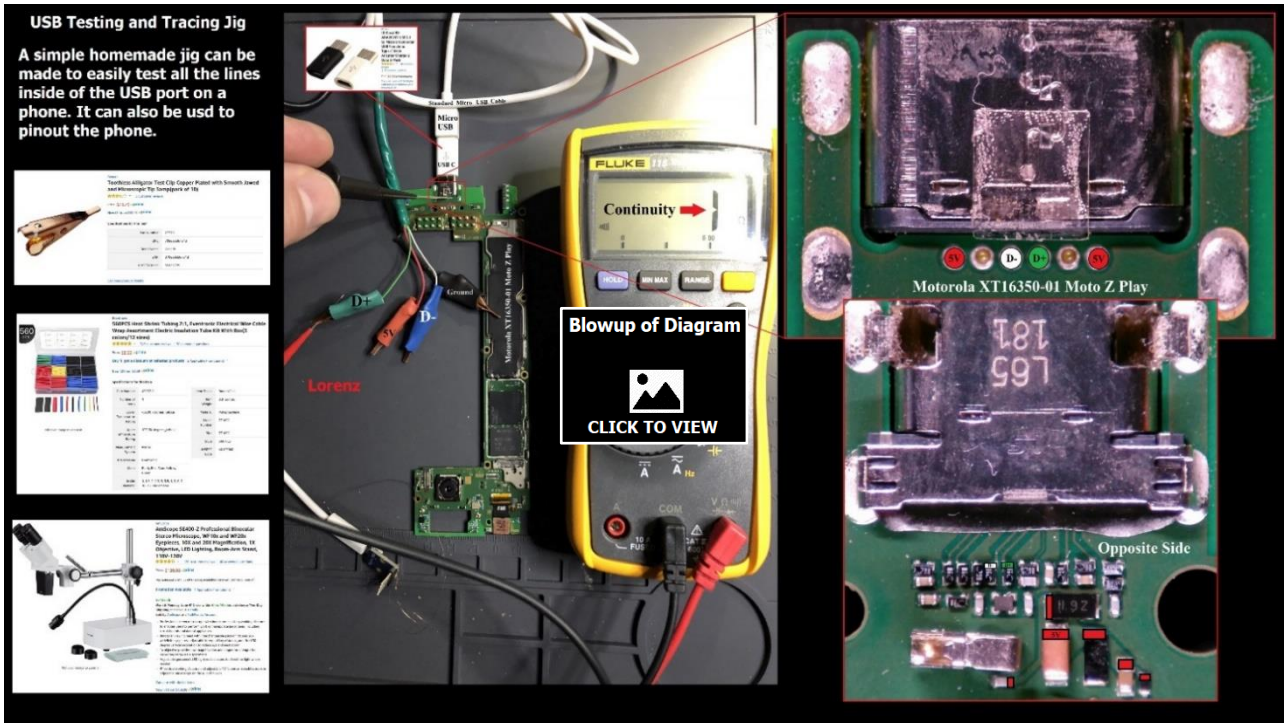
3.7.1 USB Finder Cable

In many diagrams, I mark the location of USB lines and pads and have traced them all the way back to the processor on devices. I have done that because USB ports are the gateway to the data on the phone. With few

exceptions, all extractions involve a USB port. USB ports also get the most action and wear and tear. Sometimes the only thing standing in the way of an extraction is a working USB port. For damaged devices, it may be necessary to tie into USB lines at alternate locations. There are times when extraction attempts are failing for an unknown reason and the USB port may be the culprit. It is for all these reasons I developed some simple tools to make testing and probing USB ports and lines much easier.

Ever have a phone that will charge but won't handshake with USB? Ever have a phone that appears to be dead, won't charge, or won't handshake? Before calling time of death, check the USB port. It is the source of more confusion and frustration than any other part of the phone. It is also the reason why there are phones sitting un-extracted on shelves in evidence rooms all over the world. Imagine trying to determine if a D+ line is bad by sticking a multimeter probe inside of a USB port. It's a nightmare. This simple modified USB cable works with any port just by adding an adaptor. USB ports and lines are always the same no matter if they are micro USB or USB-C. This cable will work for any port. With this modified cable and a multimeter, it is easy to see if a USB port is functioning beyond the point of where the cable meets the phone by using a multimeter and continuity. I use this cable to tell me which pads on the main logic board are the D+, D-, and 5V pads versus other pads that may be the location of the test point for which I am searching. I use this cable to trace USB lines all the way back to the processor.

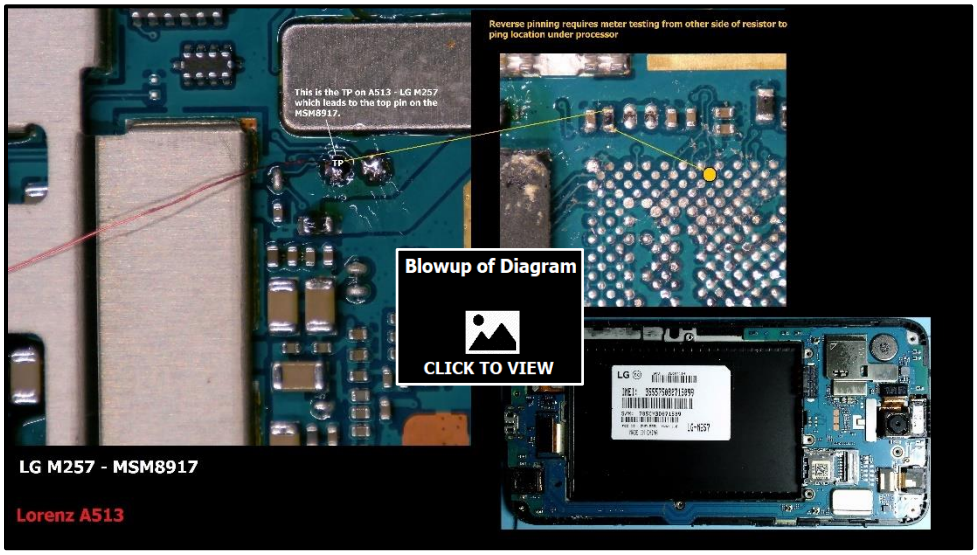
When I hook up this cable to a phone and I don't get a tone on D+, D-, or the 5v line where the port is soldered to the phone, I know that something on the inside the port is broken or damaged. I now know why everyone who handled this phone said it was broken. Its just a bad USB port. This USB finder is a lifesaver. I have several variations of it. The one in the photo shows me using clips to attach to multimeter probes. I have a USB finder with female connectors that allow me to plug in one of my multimeter probes. Either way works and anyone can make this cable with some simple tools.

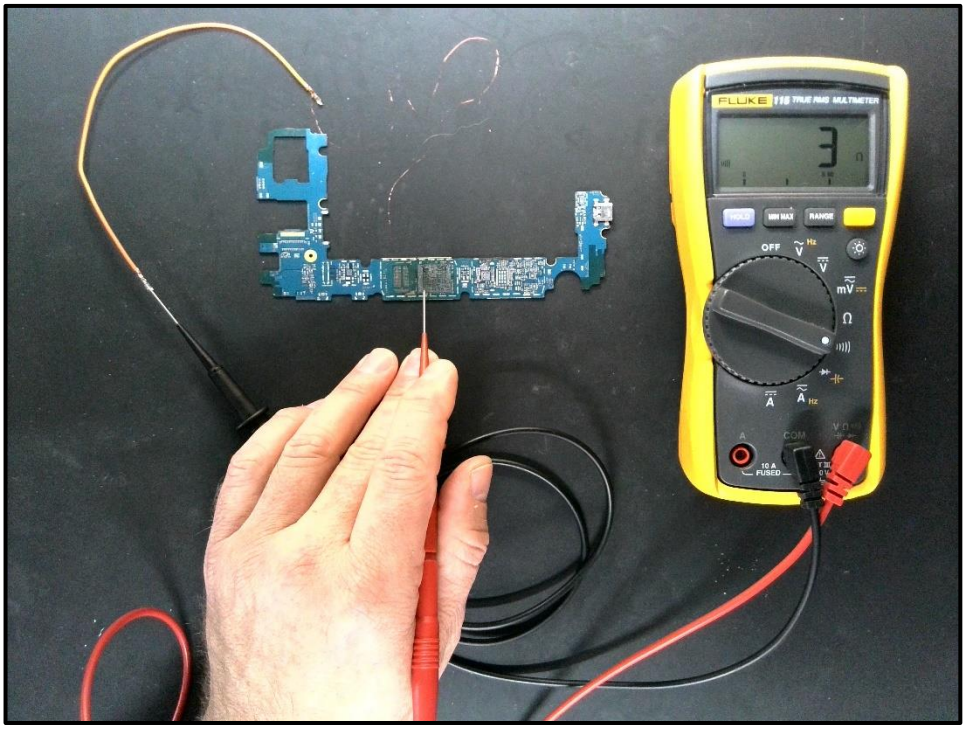


3.7.2 Pinout Jig

Many examiners with ISP experience are familiar with the concept of pinning phones. A few years ago, I started pinning processors for ISP locations and then for JTAG locations. Today I pin all over the phone, tracing USB lines, volume lines, the home button on Samsung devices, and for test points. It can be tedious to hold one probe on a known location, like the test point location under a processor, while moving the other probe all over the phone in search of a point connected by continuity. Having a simple piece of wire that I can solder to my known location and then connect to one probe makes life much easier. This is especially true when I am searching for a location on the opposite side of the board from my known source. Once I solder the wire to my known location and connect the wire to my probe, I can easily search all over the phone by just holding one single probe and have a free hand to move things around. This simple tool is what I use the most when pinning phones.

The Pinout Jig is just a piece of wire with a female connector on one end that allows me to connect my needlepoint multimeter probe. I have a very small wire I use for ISP, connected to the other side of the jig that I can solder to any known location on the logic board. Now that one probe is connected to the logic board by way of the jig, and I am free to use the other probe to search for continuity elsewhere, including on the opposite side of the logic board. I use the jig frequently for pinning processors, searching for EDL test points, and tracing lines for USB bypass.





3.7.3 Cable X

I have shorted, shocked, soldered and otherwise abused many phones during the process of testing for EDL Mode. I have many test phones and I have shorted some of them hundreds of times in every way that can be imagined. While doing this, I have damaged phones in various ways. I have replaced CLK and CMD resistors on phones and otherwise have had to repair devices I damaged just by wear and tear. I have helped other examiners diagnose and repair phones damaged by suspects and attempts to ISP or attempts to create faults for EDL Mode.

When I started researching test points, I was shorting and testing dozens of locations on a single phone. That meant disconnecting and reconnecting from USB repeatedly and applying pressure with tweezers, needles, and probes all over the phone. It was for all these reasons I developed a Cable X. I call it Cable X because I don't really know what to call it. After I made the first prototype, I realized how much easier it made everything I was doing. I made the second version after using the first one for a while.

I initially designed Cable X to do massive amounts of testing and probing while doing minimum damage to a device. After using it, I realized how well it works for any extraction and I now use this cable for just about every extraction. Cable X is designed to work by itself and to work in conjunction with any other cable used by Cellebrite. There is nothing very sophisticated about its design or how it works, although I am experimenting with some modifications for specific devices. I will include some videos on the use of the cable and I will use the cable in some demo videos on extracting devices. The following paragraphs will include some of the features of the cable.

3.7.3.1 Cable X is an extension cable

I originally designed Cable X from a standard Micro USB cable. I could use just like cable 100 or 170 for Cellebrite with the use of a Micro USB to USB-C adaptor. My second prototype is a modified 3ft USB 3.0 extension cable. This way I can attach any standard device cable, or any cable designed by Cellebrite to it. After doing months of testing methods for creating EDL Mode via eMMC/UFS faults (shorting) and test points, I found that USB 3.0 seemed to be more consistent and experienced fewer problems when shorting and probing. I was doing hundreds of USB connects and reconnects a day and applying all kinds of conditions from shorting to high voltage. It is normal to have to reboot to reset the ports after multiple USB connects and creating EDL Mode repeatedly. I found that I was having to continually restart and had more issues with voltage when using USB 2.0. When I switched to the USB 3.0 port, I have fewer issues and find that my connections and testing is more consistent. Please keep in mind that my experiences with USB 2.0 and 3.0 are not what I would consider exhaustive or conclusive research. It was primarily on one or two PCs that I use for testing and creating videos. In any case, Cable X will work the same on USB 2.0 or USB 3.0. The extension part of the cable is an Amazon 3.0 cable. Not the most expensive or premium cable on the market but I have had good luck with those cables

3.7.3.2 Cable X has a spring-loaded probe

The probe part of Cable X is a spring-loaded pushpin that I borrowed from a GPG Riff Box JTAG probing cable. I used that spring-loaded pin but tore it out of the JTAG probing switch made by GPG. I soldered that pushpin to another modified piece of a clip used to connect to batteries. This spring-loaded pushpin is designed to absorb some of the pressure created while shorting. There is nothing innovative about this as these push pins have been used for years for this purpose. Using this to create eMMC/UFS faults is a noticeable improvement. On devices I have shorted dozens of times, the damage was minimal and maintaining solid contact was much easier.

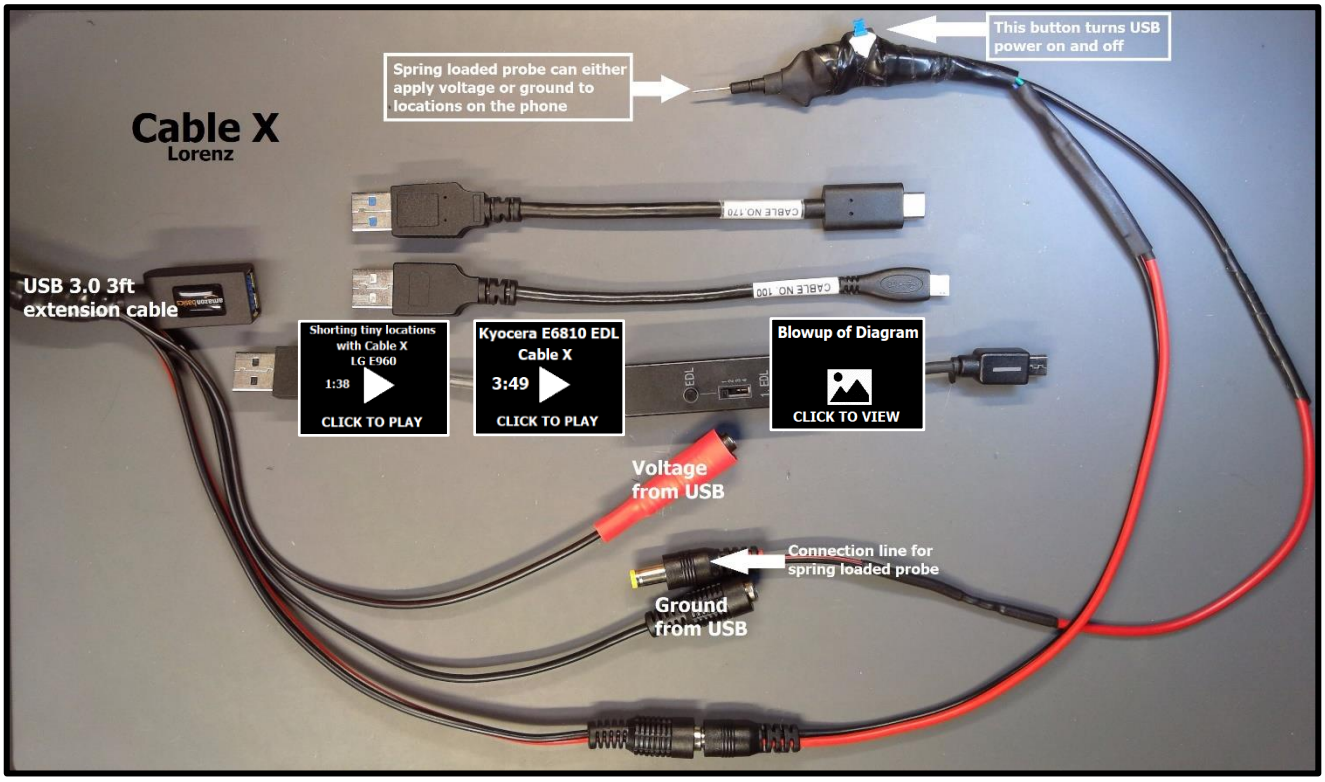
3.7.3.3 Cable X is Self-grounding

The spring-loaded probe on Cable X is grounded to the cable itself. That means once you connect Cable X to your device, the probe is grounded and ready to create the eMMC/UFS fault to create EDL Mode. No need to connect

the other end of a needle or solder a wire to ground somewhere on the phone. This makes creating faults much easier, safer, and quicker.

3.7.3.4 No need to coordinate plugging into USB while holding the shorting probe in place

One of the frustrating aspects of creating eMMC/UFS faults was plugging the device into USB while holding a needle on the tiny CMD resistor or some small location. Depending the layout of the phone and having to flip the logic board it was easier to have two people perform the procedure – one to hold needle in place and the other one to plug into USB. Some USB cables are hard to plug in and examiners have damaged phones by applying too much pressure while trying to do this dance.

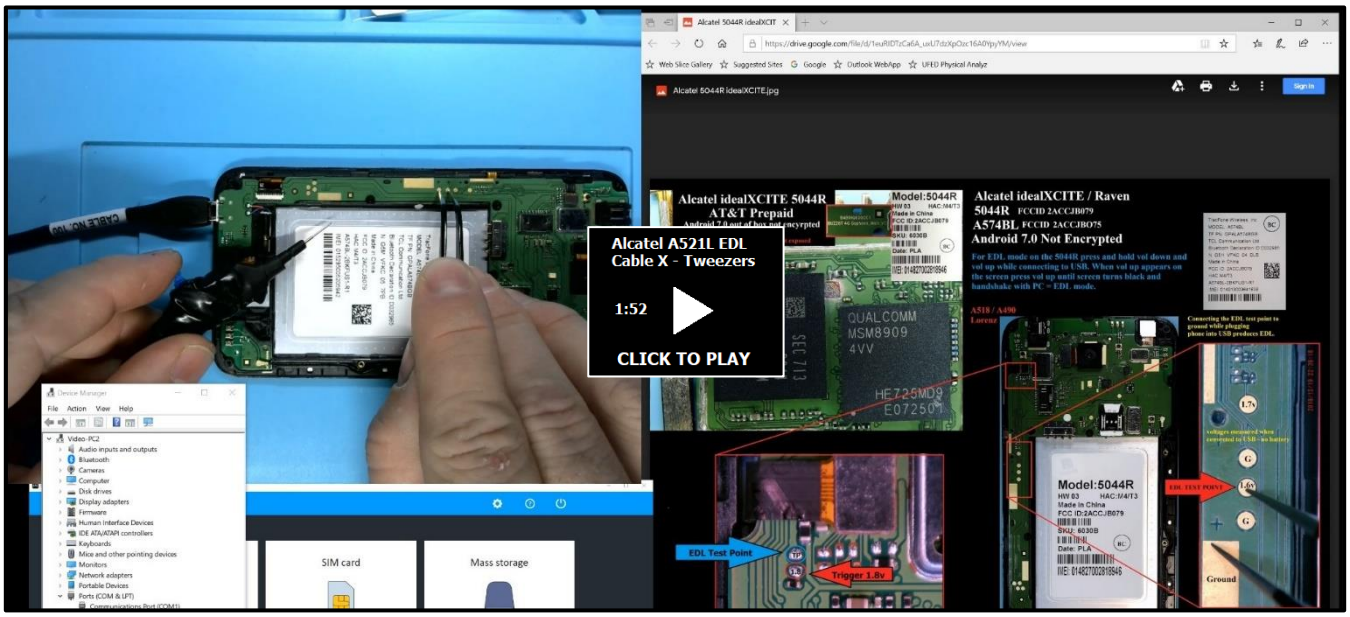


Cable X has a button that turns USB on and off. The button is located right on top of the probe for convenience, so it can be operated with one hand. Switch the USB button on Cable X off and then connect the USB cable (e.g. UFED Cable 100) to Cable X and the other end to the phone. Now touch the probe to the location you want to short and then push the button. The USB cable connects with the ground in place. You have created EDL Mode via shorting with one hand.

3.7.3.5 Cable X works well in conjunction with tweezers

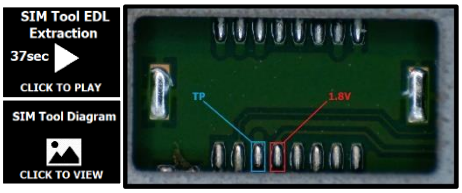
Whether you are using tweezers on an Alcatel test point that requires a connection to ground for the trigger or for any other manufacturer in which the test point is triggered by voltage, tweezers are the preferred tool for the job. That is still true if using Cable X. Even when using tweezers, the operation can be tedious on some phones and some cables and ports require significant pressure or just the correct angle to attach the cable. This is where Cable X works exceptionally well. Just use the switch on Cable X as a substitute for plugging the phone into USB. The on/off switch on Cable X allows you to connect the phone to the PC or the forensic tool before you have the tweezers in your hand. With the switch on Cable X set to OFF, you can connect the phone and then have both hands free to get the tweezers ready. Now with the tweezers in place, you can just push the button on Cable X

and the phone connects. This greatly reduces the likelihood of your hand slipping and causing damage to some resistor and it greatly improves the accuracy of creating EDL Mode on the first attempt



3.7.3.6 Using Cable X and a Motorola SIM tool for Motorola Molex test points

Some Motorola test points are located on tiny, vacant Molex pads. Tweezers are not the best tool to use on these pads because there is very little space between the pads. The pads are also covered with soft solder, which can become mashed together after repeated contact. A SIM opener tool, found in many screwdriver sets (size 0.8), works well to connect the test point to the trigger.



3.7.3.7 The probe on Cable X can be switched to voltage – Advanced Usage Only!

Most test points are triggered by voltage. If you know which point that is, you don't need the trigger to create EDL Mode. Voltage from the probe on Cable X can create that for you and you can do this with one hand with the same procedure you used to create the eMMC/UFS fault with the probe set to ground. Test points on some phones have no trigger next to them but require voltage to create EDL. Cable X can apply voltage which comes from the connection to USB. Because you can turn USB on and off with the push of a button, you control when the probe is active or not.

3.7.3.8 Caution with Cable X

Applying voltage with Cable X should be avoided unless the examiner knows something about the phone and how it works. In fact, avoid using voltage direct from the USB cable unless you have already tested the process on that exact model and know it is safe. Voltage from the cable should be used when there is no other way. Cable X applies 5 volts to the probe and this will create EDL Mode on many phones, but I am still testing this aspect of Cable X and with methods of adjusting the voltage output. However, you can use Cable X on test points without the use of the probe and it still makes creating EDL much easier. This is because you can turn USB on and off with the push of the button. Therefore, holding the tweezers on test points is much easier and less coordination is required.

4 Locating test points from scratch

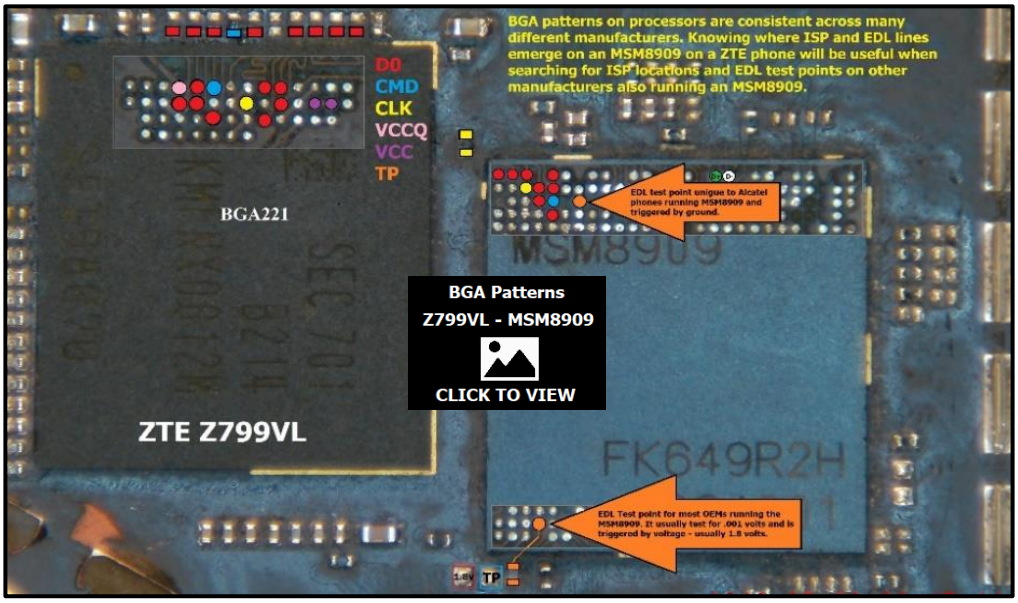
When should you attempt to locate test points? Test points should generally put you in a better situation than you are in already regarding an EDL extraction. If you have a test device you are pinning or otherwise can do with as you please, why not locate the test points on it? Even if other methods like ADB, an EDL cable, or button combos will create EDL Mode, locating test points is still a worthy endeavor. Those other methods may not always work. When I am asked why I pin an encrypted phone for ISP, my answer is, “because I’m there already”. If I am tearing apart an already worthless phone why would I not pin it just because I can’t think of a practical use for the knowledge now. It may be useful someday. If you are dealing with a piece of evidence that will likely extract via EDL Mode, and ADB, cables, and button combos won’t create EDL Mode, test points are the next step. It is how you locate those test points that will require some decisions depending on several factors. The following are generally the order of considerations and checkpoints.

4.1 Identify and confirm the presence of a Qualcomm processor

The first and sometimes the most important consideration is to confirm that the device you have has a Qualcomm processor. If it does not, there is no EDL Mode and thus no reason to search for EDL test points. This may seem obvious but what processor a phone is running is not always obvious. Some variants running the same board may be running Qualcomm processors along with MediaTek or Exynos processors. For example, Samsung Galaxy J3 phones can either be running a Qualcomm or an Exynos. The LG Stylo 4 may be running a Qualcomm SDM450 or a MediaTek MT6750V. Check databases, FCC photos, and check the UFED for suggested profiles for your model. If the UFED suggest an EDL extraction as a possibility, it’s running a Qualcomm processor.

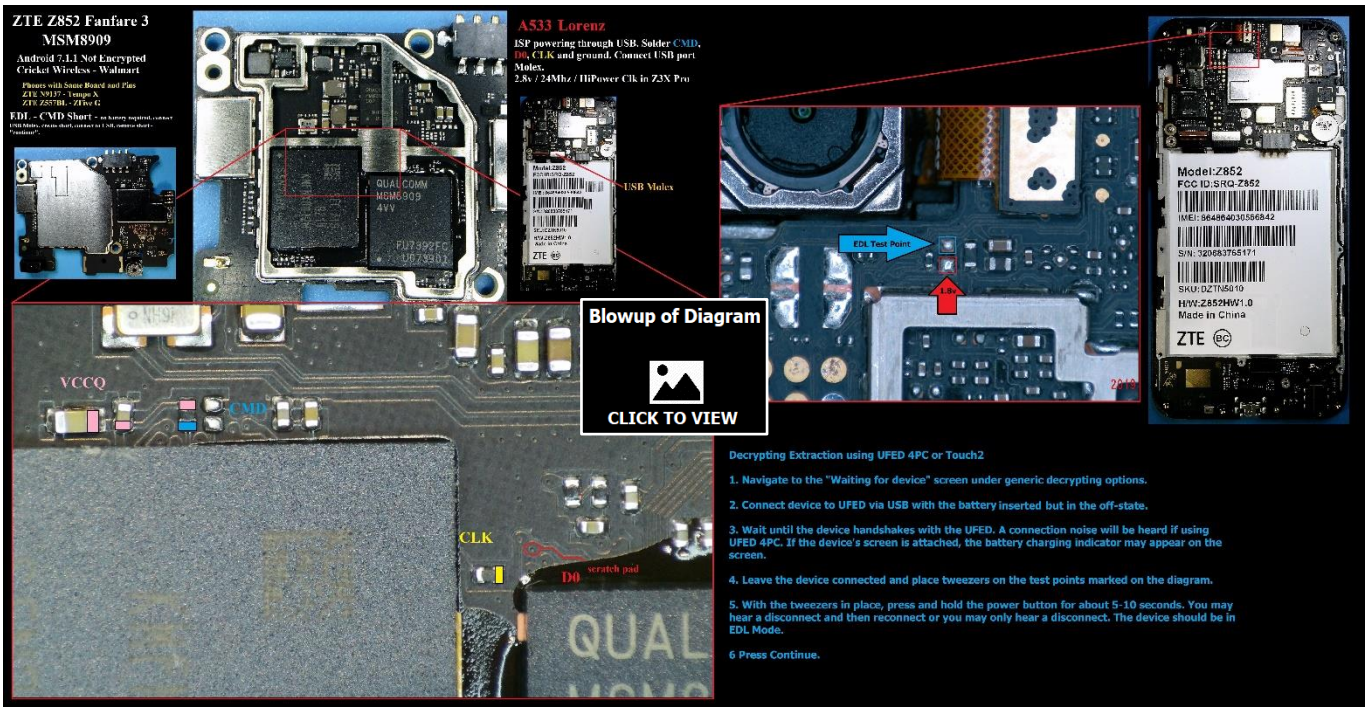
4.2 Identification of past patterns of the manufacturer of the device

Check for pinouts and test points on other devices from the same manufacturer. LG phones are easy as they all have crosses for EDL test points. Some phones with round pads that look like everything else and will likely have patterns for where and how they are located. Where are they normally located on the logic board? Are the test point and the trigger isolated from other pads? Just like identifying ISP locations or JTAG taps, manufacturers follow patterns and don’t change designs if there is no need. Seeing several examples of other pinouts from that same OEM can be a huge help when trying to narrow your search.



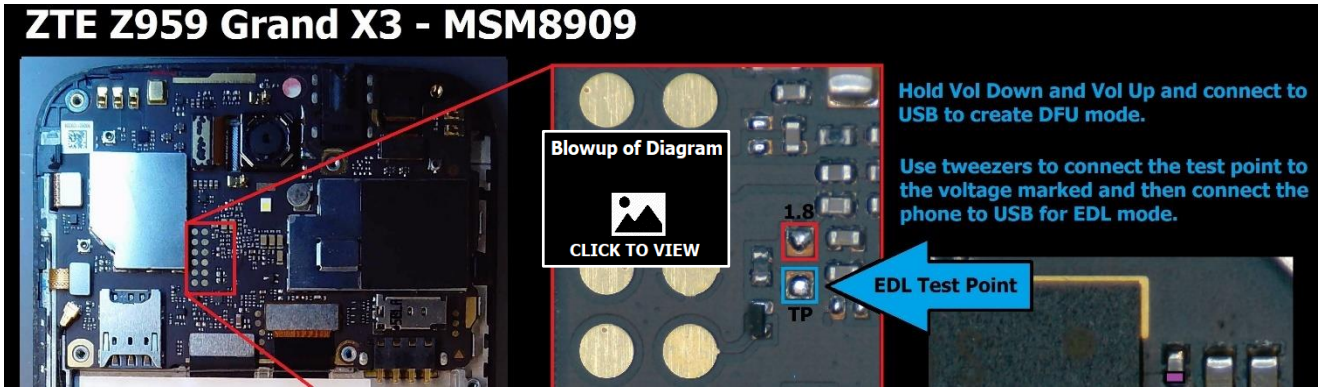
4.3 Check for device variants already marked

Device variants means phones from different carriers running the same logic board and hardware. Although these phones can behave differently regarding extraction support, EDL Mode and EDL test points and faults will generally be the same. In other words, if one of them has EDL test points they likely all will because the boards are identical. The ZTE Z852 Fanfare 3, ZTE Z9137 Tempo X, and the ZTE Z557BL ZFive G all have the same logic board and layout and the test point is the same on all of them. Their model numbers and nicknames are not similar but they all behave alike regarding EDL and are virtually identical regarding board layout.



4.4 Identification of suspected test point and trigger locations by sight

I wish all OEM were like LG regarding test points. There are only two pads shaped like crosses on modern LG phones. One of them is the EDL test point and the other one, right next to it, is the trigger. No probing or guessing necessary. Get a pair of tweezers and you are in business. There are other OEMs that have test points which can be recognized by sight, especially when you also consider their location. ZTE phones have round pads as test points that look like everything else on the phone, but some have square pads covered with solder that are recognizable and stand out. Seeing many diagrams side by side from the same OEM is a good way to become familiar with patterns.

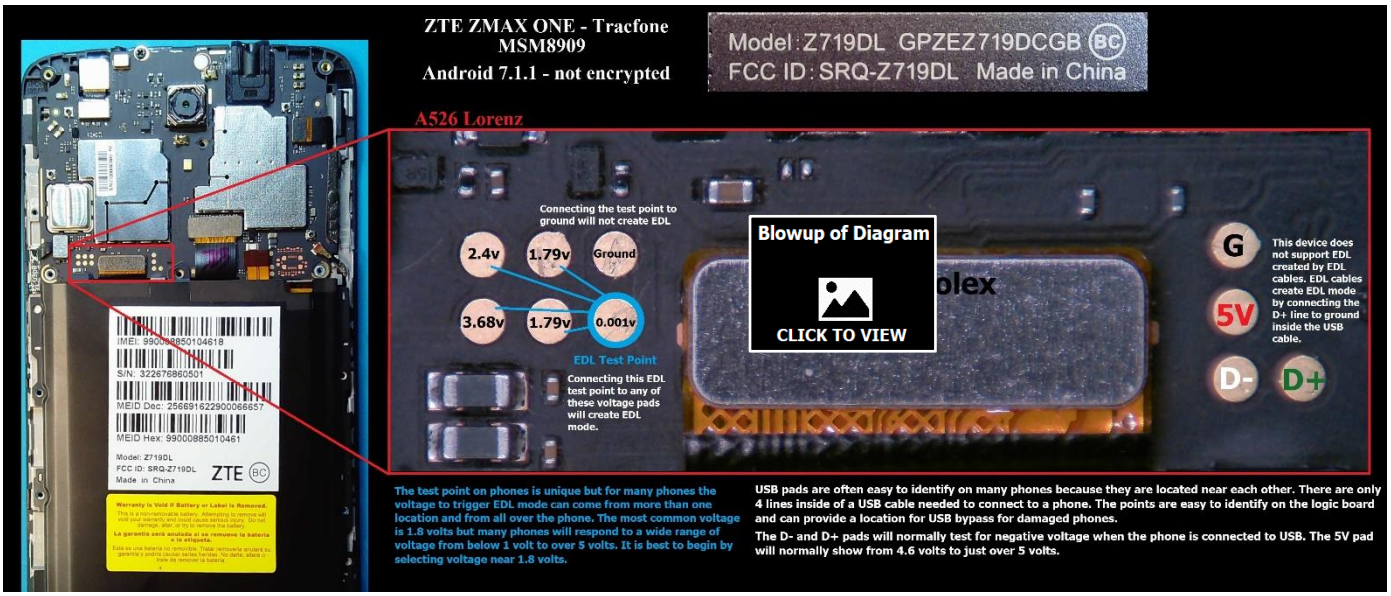


4.5 Use of the multimeter to minimize time and maximize probing efficiency

Probing for EDL test points is the act of triggering a suspected location by applying a ground or voltage to see if EDL is created. The idea behind researching, looking at patterns, and the use of a multimeter is to limit the amount of probing you must do. Ideally, the first attempt creates EDL Mode. After researching this, I have been able to pick the EDL test point on the first try after using all the above tactics and then confirming with a multimeter. When two pads are next to each other in a conspicuous location and one test for .001 volts and the other test for 1.79 volts, I know I have two good suspects for a test point and its trigger. Of course, if it's an Alcatel phone the test point is voltage and the trigger is ground. When I come across a phone with many pads next to each other like some ZTE phones, I know I will need my multimeter to help reduce my number of attempts at probing. You can draw the circles on a piece of paper and list the voltage on each and then pick your suspects for test points and triggers. Then make your attempts.

4.5.1 Multimeter probing the ZTE Z719DL for the test point

The ZTE ZMAX ONE, Z719DL, is a good example of using a multimeter to test for a suspected test point. When I tested this phone for the test point, it was easy to narrow the choice so that my first guess was correct. I know the test point on ZTE phones is always triggered by voltage and not ground. I know the test point usually tests for very



low voltage when the phone is connected to USB power with no battery. I know test points are normally triggered by 1.8 volts. I know that D+ and D- pads (USB data lines) test for negative voltage when the phone is connected to USB. Ground pads are usually easy to identify visually without a meter but if you do probe a ground location it will show all zeros immediately. Test points will vary but many will test out at 0.001 volts with the phone connected to USB power. Probing for test points is not always as easy as the Z719DL, but using a multimeter is indispensable when probing for test points. Connecting this test point to ground will not create EDL Mode. The widely held belief that all test points are triggered by ground is one reason why test points on many phones have gone unnoticed.

4.5.2 How to power the phone when testing with a multimeter

The most common method I use is to disconnect or remove the battery and apply USB power to test suspected pads and locations with a multimeter. Most devices will distribute power to those locations without the battery or without the phone being in the on state. Some phones will repeatedly attempt to boot when USB power is applied with no battery. Many Alcatel phones do this. It doesn't hurt to test points for voltage during these attempted booting cycles. Just keep your voltage probe on the suspected location through the cycle. Your meter will display nothing and then ramp up to the normal voltage of that location and power down again. The maximum voltage you see is what is consistently there if there were a battery in place.

4.5.3 Some phones require the battery inserted for voltage testing

Some phones will require the battery to see the voltage of a pad or location. Some of the older Samsung phones running the MSM8916 are like that. If you remove the battery and just connect to USB on some Samsung phones, many locations will appear dead when you test with the multimeter. Usually inserting the battery will make everything light up, even if the phone is in the off state. For example, pads that showed very little or no voltage when connected to USB with no battery will now show 1.8 volts. I have been able to get voltage to appear on some Samsung phones by holding the power button with USB connected but the better way is for the battery to be inserted. For those with experience with JTAG, you know that some phones require the battery to be inserted or for the power button to be depressed, or both, to establish a JTAG connection and detect the "dead body" of the phone. The same is true for some phones to test for voltage and to create EDL Mode. If you need the battery to test for voltage on the phone, you will likely need the battery inserted to create EDL Mode using the test point but not always.

4.5.4 Placing the phone in EDL Mode to look for a test point to create EDL Mode

I like a phone in EDL Mode. It is very peaceful and compliant. Ironically, once EDL Mode is created, everything is stable, and the phone has voltage at all locations. Including the test point trigger that creates EDL. Sometimes you don't see the true voltage of a trigger until EDL Mode is achieved. That makes some triggers hard to spot as the voltage is low until the moment of EDL. I have used triggers that tested for 0.6 volts with USB and no battery and when that trigger was connected to the test point, it created EDL and the voltage for the trigger then stabilized at 1.8 volts. On devices I am testing and pinning, I can sometimes easily spot the CMD or CLK location and I will just short the phone into EDL Mode. With the phone in EDL Mode, all the other pads and locations have stable maximum voltage and I get a true picture of the voltage of each location. I can then find my suspected test point and triggers, remove power, reset the phone, and test my suspects.

4.5.5 Why bother finding the test point after you have used a fault to create EDL Mode?

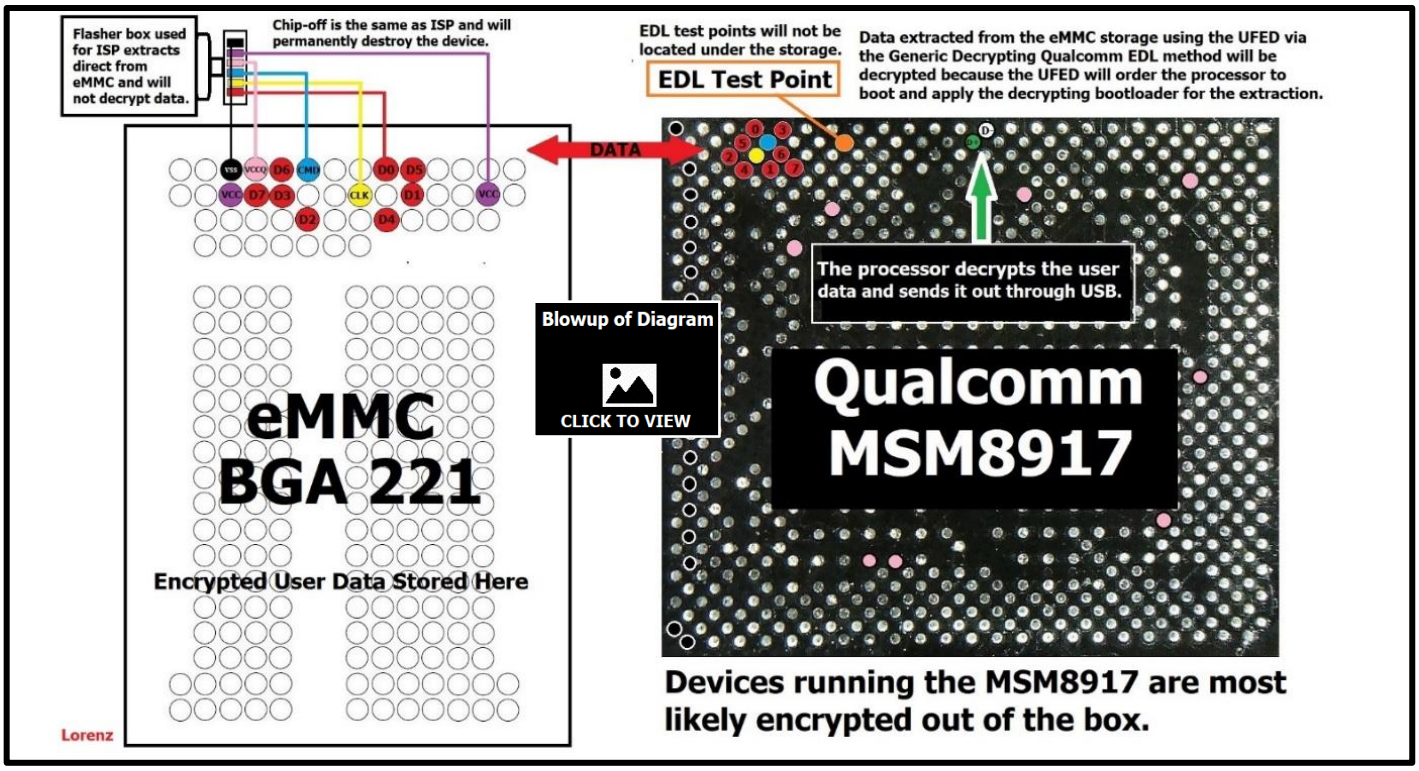
Finding the test point on a device will also provide me the location of the test point on the variants of this device. That test point will likely be easier to reach than a fault location and thus save time and teardown when I receive this device's brother. It will also mean that when I receive this same device again, I can go right to the test point and possibly save removing a tough heat shield. Of course, the obvious reason to pin phones you have already extracted is to share that information with other examiners.

4.6 Last resort – chip the processor on a test phone

When I first started researching test points, I used phones with known and proven test points to locate the specific point on the BGA of Qualcomm processors for this very purpose. Before I became familiar with all the patterns and characteristics of test points, I chipped the processor on phones with unknown test points if they were running a Qualcomm I had already pinned for a test point. I am still not finished researching and pinning processors. I have pinned over 100 processors for ISP, JTAG, USB points, EDL test points, and various other points of interest. Some processors I pinned years ago I am going back and repinning for EDL and other points.

The EDL test point is unique but the trigger can be anything. EDL test points cannot be located by chipping the storage on phones. To pin a phone for ISP, you can chip the eMMC and trace the known points to the ISP locations. You can do the same with the processor because D0-D7, CMD, and CLK connect the processor to the storage. However, the EDL test point is not connected to the storage; it is connected to the processor. The same is true of USB data lines. So while the BGA under the eMMC can only be used to look for ISP points, the BGA under processor can be used to find everything; everything except VCC that powers the eMMC. Very few processors have a shared VCC pin with the storage.

Processors are pinned for test points but not triggers. I do mark processors for VCCQ for ISP purposes. Also note that just because a pad test for 1.8 volts, that doesn't mean it is the VCCQ needed to power the storage for an ISP extraction. Even though 1.8 volts is the common trigger for EDL test points, pads and points testing for 1.8 volts are common on phones, so just the presence of that voltage doesn't always mean you are near a test point.



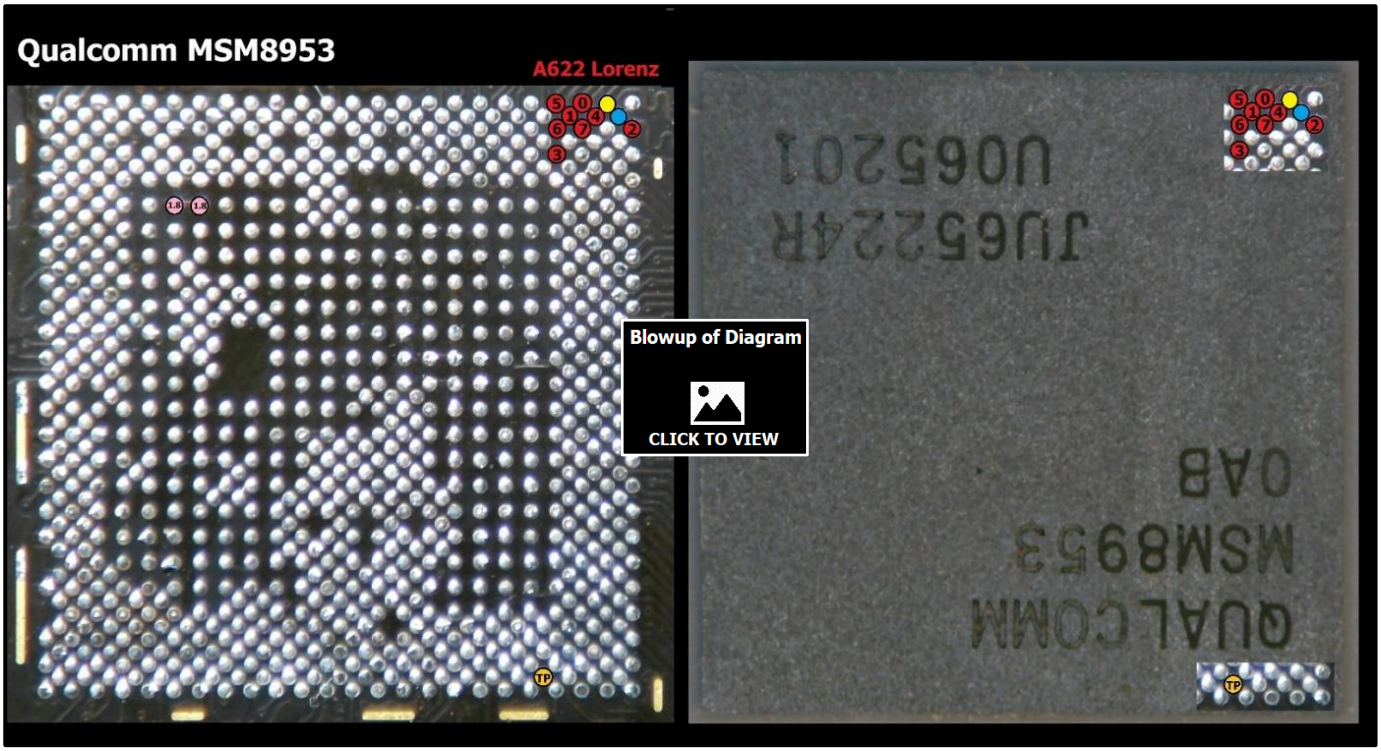
4.6.1 Chip-Off orientation – matching the diagram to the chipped board

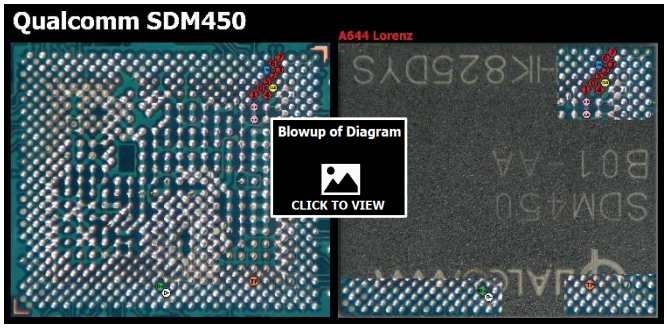
Most processors usually have a BGA pattern that makes it easy to determine which end is up when looking at a logic board where the processor was removed. If the BGA was one uniform square of pins with the same number of pins on each row, it would be difficult to determine which pin is the one you are looking for on a processor pinout. For those familiar with pinning phones for ISP and chip-off, the orientation pin on the BGA153 eMMC, is

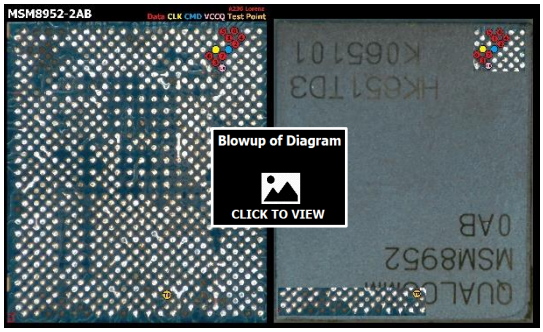
necessary in order to know which pads are which, on an otherwise perfectly square chip with no distinguishing pattern to identify top or bottom.

4.6.2 Processor orientation – knowing where ISP and EDL test point lines emerge

Processor orientation also refers to which side of the processor EDL or ISP lines emerge. Knowing this information can help examiners needing an ISP location or a test point to create EDL Mode, when other methods have failed. Yet again, knowledge of ISP and Chip-Off is of use in this situation. Most processors are oriented so that the ISP lines emerge from the top of the processor. By top of a processor, I am referring to the top in relation to the labeling of the processor. A few processors, like the MSM8952, MSM8953 and SDM450, are oriented upside down. The ISP lines emerge from the bottom of these processors. The diagram of the MSM8953 is an example designed so that you can see through the top of the processor to the BGA pins beneath. Note the test point is located at the top of the MSM8953

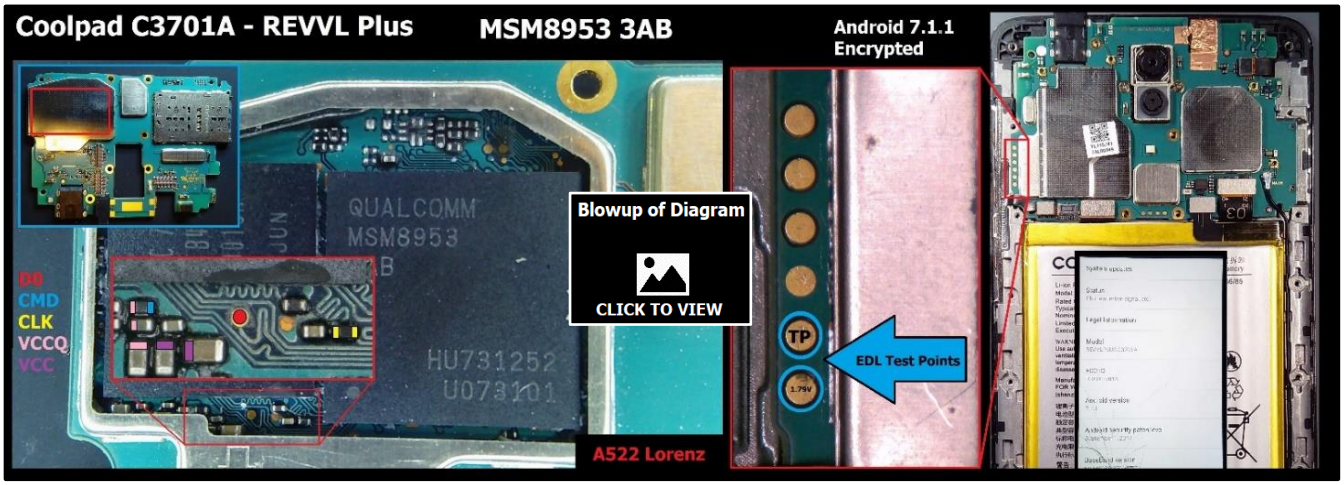






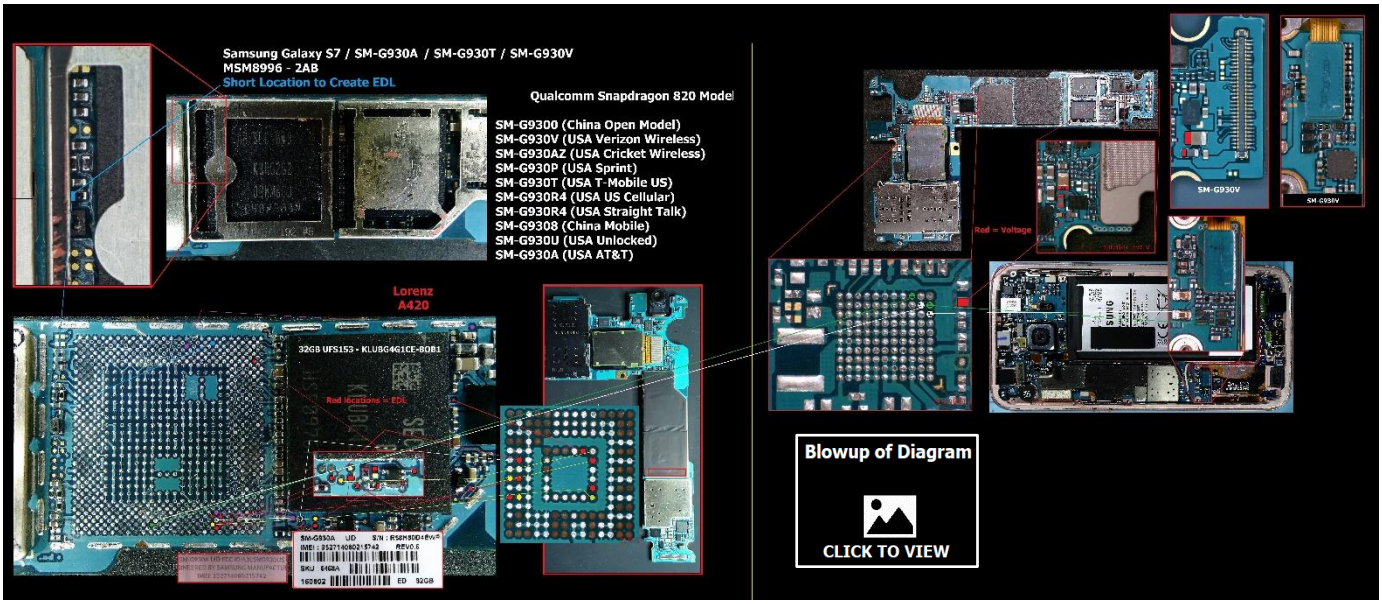
4.6.3 Coolpad C3701A – REVVL Plus - MSM8953 orientation

Knowing what is under the MSM8953, helps find ISP points as potential short locations for the Coolpad C3701A. The diagram of the Coolpad C3701A REVVL Plus reveals the ISP points located at the bottom of the processor.



4.7 Pinning processors for various points of interest or tracing a point to the processor

I wanted to sum up with a final word on pinning the processor for points of interest like test points, USB points, home buttons, or voltage, or going from the processor to those points in reverse. As I mentioned earlier in this paper, the path from the processor to the test point or other locations may be broken completely by a switch or attenuated by resistors. If the path is broken it is typically by a power regulator in which the line goes in at one point on the BGA of the chip and comes out another BGA point on the chip. This can make it challenging to trace some USB lines and voltage directly back to the processor, even if you remove that power regulator on a test phone. It can be frustrating and tedious with some phones. I have done a lot of exploratory on some test phones trying to find ways to boot phones with catastrophic damage. Sometimes locating a test point via chipping is easy, but sometimes it is very frustrating.



5 Extracting devices with Cellebrite's UFED

Once you have established a way to get the phone into EDL Mode extraction is the next step. Of course, you must accept the possibility the extraction will fail or may not be supported. This is another reminder that EDL Mode is a creation of Qualcomm and they didn't put it there to help forensic examiners get into devices. The ability to exploit EDL Mode for extracting a device is limited to certain processors and devices. There are decisions to make regarding how to begin an EDL extraction, beyond getting the device in EDL Mode. This section will cover strategies and considerations for determining the best course of action using the UFED, but they are also helpful for other tools as well.

5.1 Why do I always use and write about the use of Cellebrite and the UFED for EDL extractions?

I didn't want to make any part of this document personal, but I also believe it is important for readers to know why or why not when it comes to forensic tools I own, use, or write about. I don't have time to test every tool available or every tool I have for every device I cover in documents like this. There is an extraordinary amount of time involved in testing, probing, and pinning phones. Making videos and diagrams of extraction flows is extremely time consuming.

I am a full-time professor and member of the faculty at a local college. I also perform digital forensic examinations on computers and mobile devices, mostly for law enforcement agencies. I do that as I am needed, which usually works out to be another full-time job. My research and experimentation with mobile devices are part of my profession but it takes a great deal of time to test, document, and validate devices that are the subject of the material I write. Fortunately, all my work on research and writing is related to teaching and forensic examinations for criminal cases so one task benefits another.

Like many examiners and agencies, I own multiple forensic tools including Cellebrite, EnCase, Oxygen, and Axiom, and I have utilized every tool I have at one time or another. I also own and use Z3X, Z3X Plus, Riff 1 & 2, ATF, Medusa Pro, NCK Dongle, eMMC Pro, and Chimera. However, I don't have time to conduct a thorough or fair comparison or point by point analysis comparing every aspect of one tool to another with every tool I have or other tools which I have not listed and do not own. I therefore do not intentionally try to push one tool over another. I advocate owning and using multiple tools. If money and time were not a factor and I would own and test them all. Cellebrite just happens to have the widest support for the universe of devices and the UFED can do things with EDL extractions that other tools cannot.

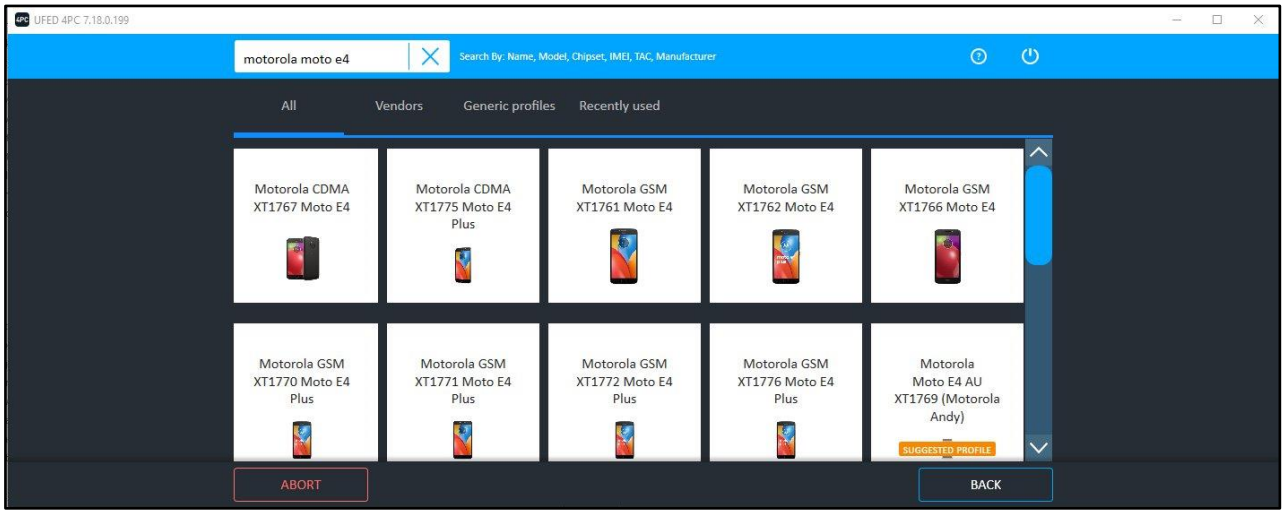
Therefore, my discussion of extraction techniques and considerations will involve the use of Cellebrite's UFED. For those who use other software or tools for EDL extractions, the information here will assist you no matter what method or tool you use. I spend a good deal of time talking about ISP, JTAG, and Chip-Off and I frequently use those techniques and tools for the work and research I do related to EDL extractions and research. I strongly believe training and practice with ISP, JTAG, and Chip-Off is extremely beneficial with all extractions and particularly with EDL extractions. I highly recommend that training for any examiner. With forensics, the more tools, techniques, training and options you have, the better.

5.2 Researching the universe of phones: The UFED as a database

I have my own database I developed for research, organization, and a way to track my inventory of test devices. I built that database based on the UFED Supported Phone List and I use Cellebrite's release of supported phones as one method to stay up to date on new phones hitting the market. I have added devices and variants not specifically listed in Cellebrite's spreadsheet at the time. Cellebrite releases a spreadsheet of supported devices (e.g. UFED_Supported_Phone_List_7.18) available for download and usually updated when new versions of the UFED are released. When that updated list comes out, I can quickly determine which are newly added devices simply by filtering the "Date Added" column for phones not present in my database and newly added to the UFED.

I wrote about using Cellebrite's supported phone list to help determine whether a device is likely encrypted or not in Mastering EDL Mode. See the full section of Mastering EDL Mode on Determining device encryption.

Even if you never use the spreadsheet, the UFED Touch2 or the UFED 4PC has an excellent built-in database of devices, which can be searched by nickname or model number. The universe of devices narrows as you type so that even when you are not sure exactly what you are looking for, the UFED can guide you. Cellebrite has a very comprehensive list of both smart phones and feature phones so the UFED's database benefits me even if I am going to use ISP or JTAG to extract a device. I use the UFED in combination with online databases to determine extraction possibilities, processor identification and support, and identification of device variants.



5.3 EDL support based on specific processors in the UFED: Widely Supported

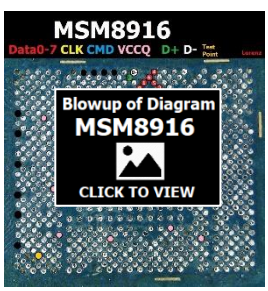
With the UFED's comprehensive support for selected processors, I know that devices running that processor are almost a sure bet when it comes to EDL extraction, even if they are locked and encrypted. Devices running processors with limited support are also worth attempting an EDL extraction but there are considerations for attempting extractions on devices with limited support. Instructions in the UFED for supported devices tells me if the EDL cable or button combinations will work for extractions.

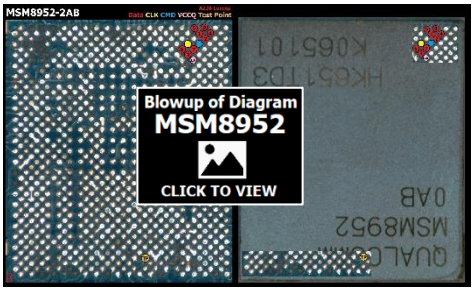
5.3.1 UFED Widely Supported Qualcomm Processors for EDL extractions

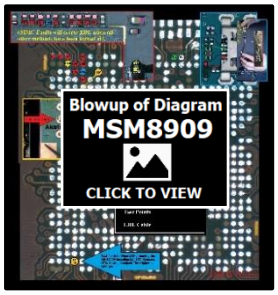
Cellebrite Supports EDL extractions on any phone running these processors.

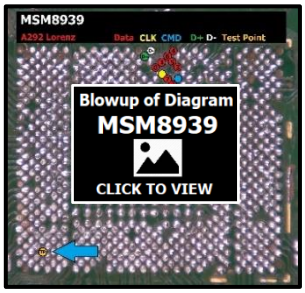
MSM8909 MSM8916 MSM8936 MSM8939 MSM8952

"Widely Supported" is a term used by Cellebrite to denote five Qualcomm processors in which they can always exploit via EDL. If a device is running one of these processors, an EDL extraction will work, whether locked or encrypted. There are rare exceptions which can normally be remedied via tech support. Of course, Secure Startup









will always require CAS. Secure Startup is still confusing for some examiners regarding what is possible and how it effects EDL extraction. There is a detailed section on Secure Startup in Mastering EDL Mode that explains it in detail. Both "Widely Supported" and "Limited Supported" devices running Qualcomm processors can be encrypted either by default or by the user.

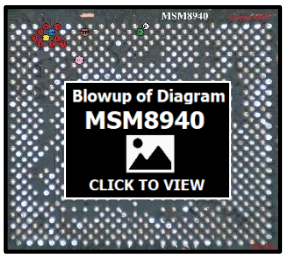
5.3.2 UFED Limited Supported Qualcomm Processors for EDL extractions

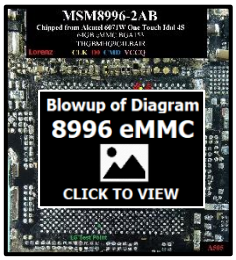
Cellebrite Supports EDL extractions on select devices running these processors.

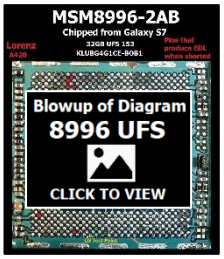
MSM8917 MSM8937 MSM8940 MSM8953 MSM8996

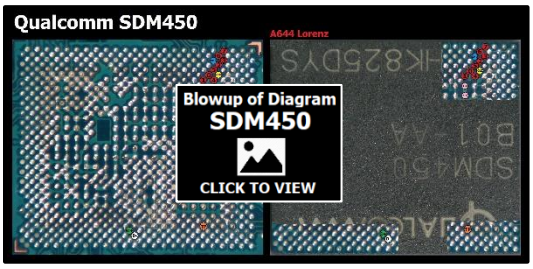
"Limited Support" refers to devices running specific Qualcomm processors which may or may not extract via EDL. There is generally no harm attempting EDL extractions as failures do not harm the device. Whether or not an attempt should be made depends on the probability of success versus what method is required to get the device in EDL Mode. In other words, if you have a device that is not directly supported for a lock bypass, physical extraction under its profile in the UFED, but it is running one of these processors below, it may be worth attempting an EDL extraction. Generally, someone on the forums will have an idea whether it's worth a shot or not or has likely attempted an EDL extraction on one of these phones that failed. Cellebrite will suggest an attempt when a device fits this criterion.

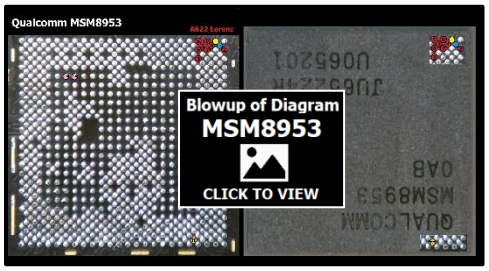
Some devices are tough to disassemble. I can get the Galaxy S8 and Galaxy S9 in EDL Mode using test points. However, the back cover must be removed with heat to get to those points and right now, there is no EDL support for those devices. Therefore, I would tell you not to try until that changes.

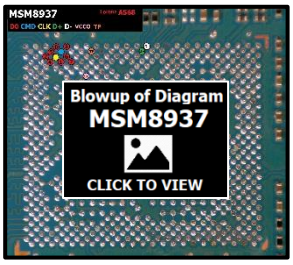












5.4 UFED's ability to convert FTM, DFU, and LAF into EDL Mode

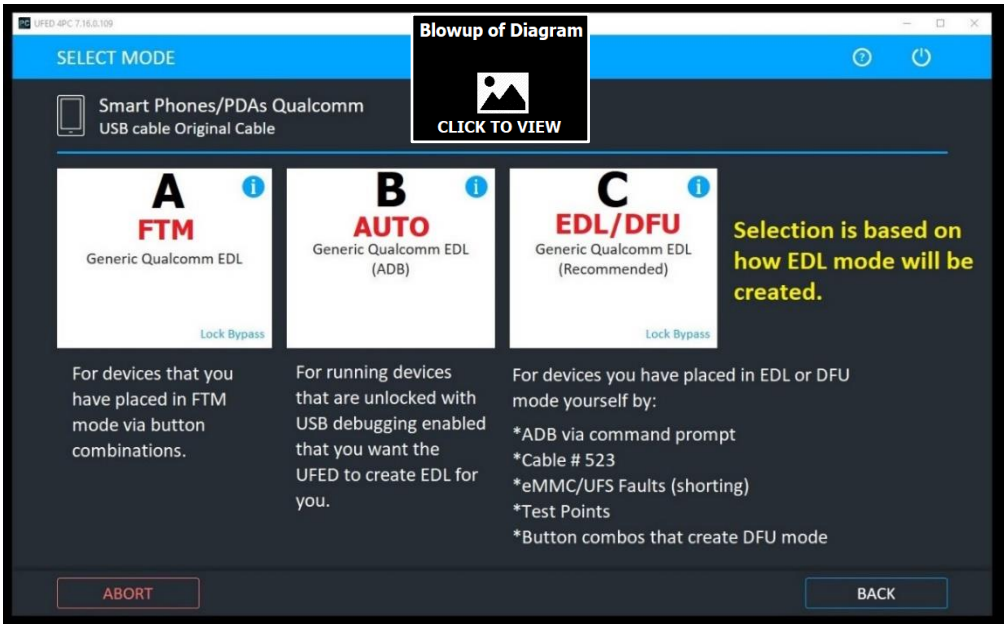
The UFED opens the door to more options to create EDL Mode by using other device-specific modes to create EDL Mode. More options for creating EDL Mode inside the UFED means EDL Mode can be created on locked devices even if the EDL cable doesn't work and if button combinations produce something other that EDL Mode. All of that means being able to avoid disassembly of devices to attempt shorting or locating EDL test points. This is also why it is important to know what button combinations actually do to the phone as extraction options vary depending on what mode the button combination created.

5.5 Many forensic examiners have used EDL in the UFED to extract devices without knowing it

In speaking with forensic instructors and examiners all over the world, I realized many examiners have used EDL Mode in the UFED without knowing it. The UFED offers physical extraction methods under specific device profiles and provides the examiner with specific instructions for the extractions. Such instruction may include the use of button combinations but may not mention EDL during the instructions or the process. Examiners may not be aware that what they are actually doing is placing the device in EDL Mode. Examiners also may not know that some button combinations place a device in DFU Mode and once in DFU Mode the UFED can convert that to EDL.

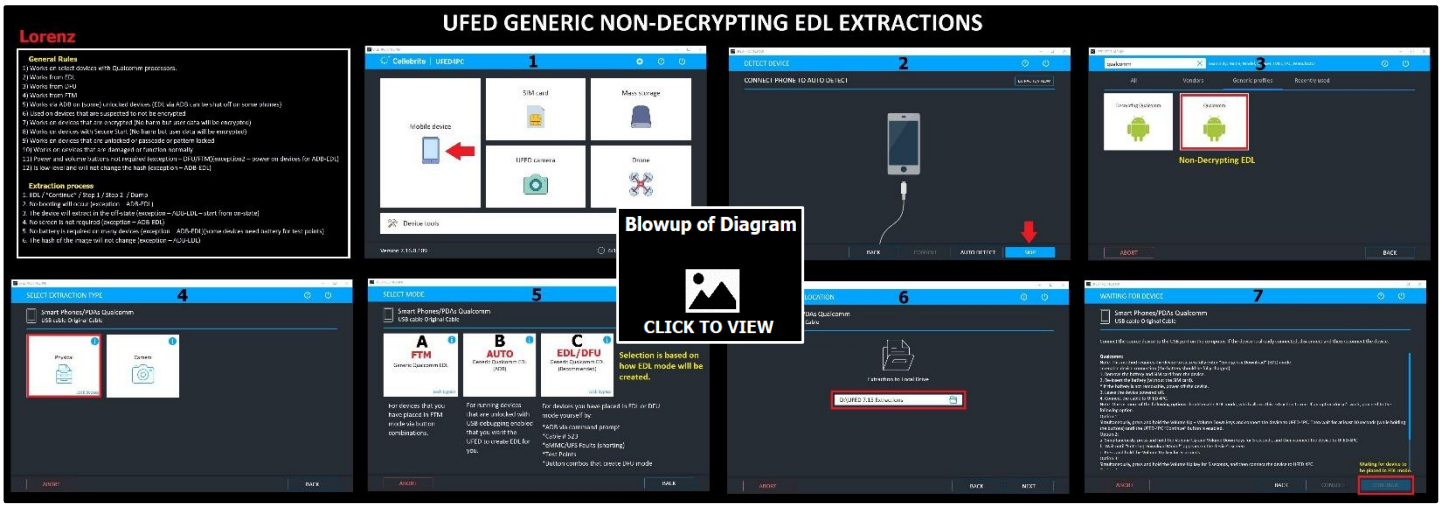
The ability of the UFED to seamlessly convert DFU, FTM, and LAF into an EDL extraction is something examiners may take for granted or assume EDL is not at play during an extraction when it is. When using the UFED, the specific device supported profiles are the preferred method of extraction for a specific device, because it means Cellebrite has tested the device and determined the optimized procedure. However, even if a specific device profile doesn't exist as a UFED profile, it may still be possible to extract that same device under Generic options in the UFED. Those generic options include the ability to create EDL through DFU, FTM, and ADB. Using the UFED to convert LAF to EDL is something this is only done on specific phones under their device profile and is not available through generic options.

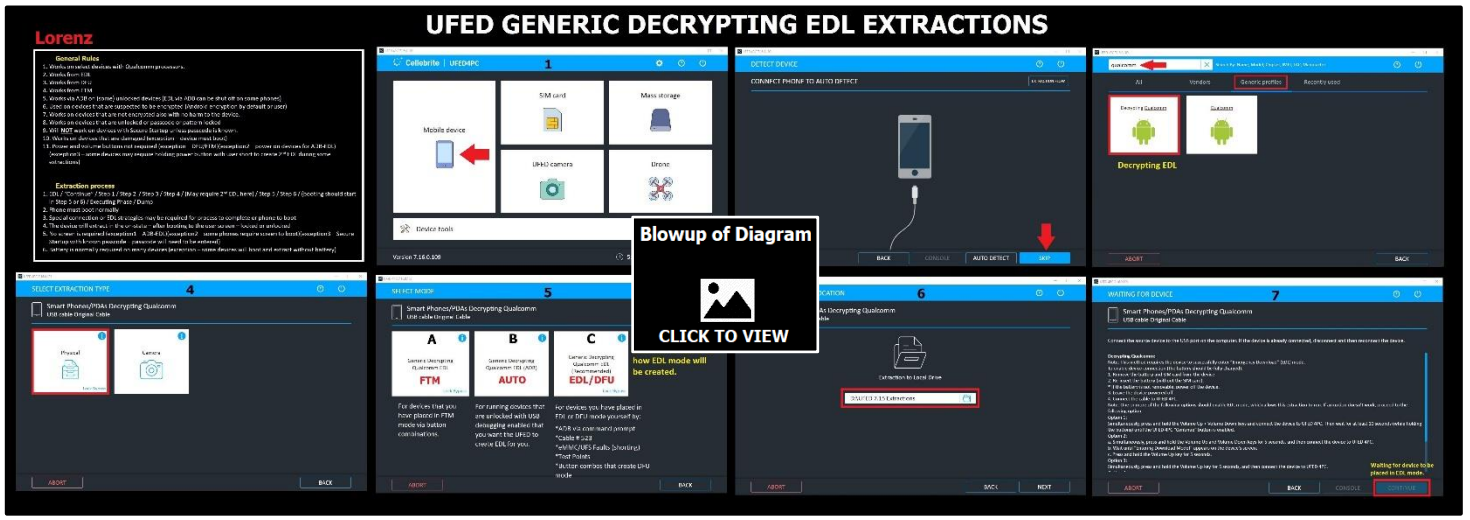
Thus, there is a benefit to knowing more about what the button combinations on a device are doing. Sometimes they create EDL and examiners just don't realize it. Sometimes the button combos create DFU and examiners assume it is EDL. It is usually apparent to the user if the device is in FTM, provided there is a working screen on the device being extracted. That is because FTM is displayed on the device screen. FTM does require an alternate mode selection in the UFED (option A) when using generic EDL options. It is important to consider which mode is appropriate if you are using ADB. There is a difference between you allowing the UFED to create EDL on a running, unlocked device via ADB (option B) versus you using command prompt to place the device in EDL Mode (option C). Note that I have created labels, letter designations, and some cheat notes for these options just of ease of discussion. The three mode options will appear whether selecting decrypting or non-decrypting extractions.



5.6 UFED's Generic Profiles and Suggested Profiles for device variants and unknowns

The UFED's "**Generic Profiles**" allows a user to attempt extractions on devices not specifically supported under that device profile but which may extract generically because the device shares certain characteristics with other supported models. "**Suggested Profile**" options allow users to attempt specific extractions which may or may not work but are a suggested possible option because of known characteristics or because the device is a variant of a tested model directly supported. The suggested profile is based on device characteristics like the brand and model of processor, which means a specific extraction or unlock may work. The universe of phones and variants is vast, and it is impossible for any company to test every phone on the planet. This is where researching device characteristics can provide examiners with some options for potential extractions not specifically listed. Many EDL extractions start in the same way – the device is not specifically listed but is running a processor supported for EDL extractions on other devices. It may not work every time but EDL extractions and attempts are generally very safe. Cellebrite can obtain decrypted extractions from devices and gives the examiner the choice under generic options. I have made a flow diagram of each choice with some considerations and requirements listed for each type of extraction.





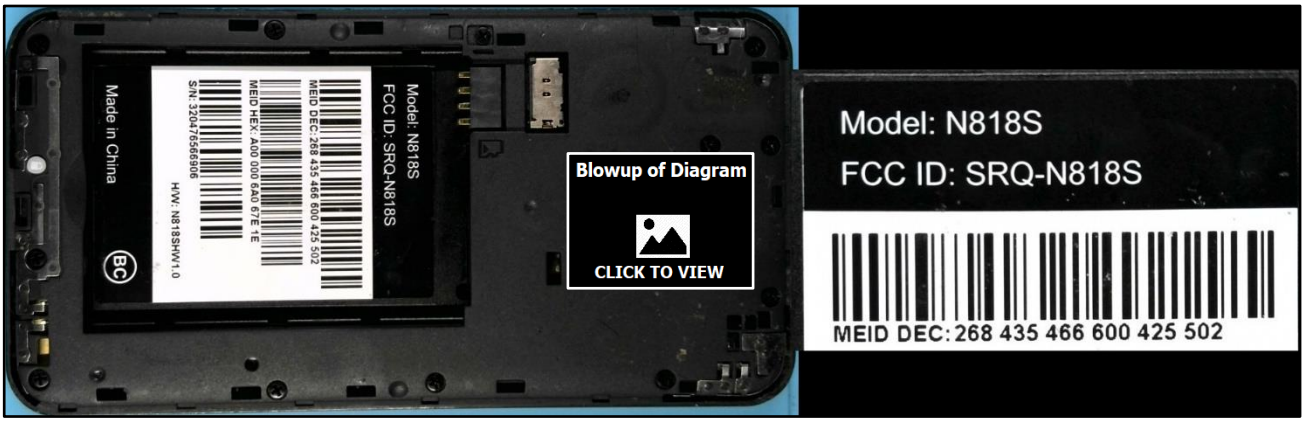
5.6.1 Using Generic options in the UFED on a device not listed

The generic options in the UFED is a powerful tool for extracting devices that may not be listed as supported under the specific device profile. Because the options are "generic", it means they are intended to work on categories of devices instead of being created for a specific device. Some generic tools involve a specific OEM running a specific processor.

The UFED Extraction Flow PDF can be found under my.cellebrite.com and is now built into the UFED in release 7.18. The Extraction Flow provides a visual of the Generic options and tools in the UFED, including processors supported for EDL extractions. It is useful when you come across a device that is not listed in the UFED. The ZTE N818S is not found in the UFED so the first step is to research the phone to locate information to help determine an appropriate generic extraction, if one exists. There are many online databases that include information on specific devices but sometimes the most commonly used databases like phonescoop.com or gsmarena.com may not have the specific device listed either. Many times, I start out with a simple Google search for the device. For this device just type in "zte n818s specs". Check the results to see what information you can learn about the device. The most important piece of information for a device like this is the processor.

The N818S is running and MSM8909 processor. Checking the Extraction Flow guide reveals that the MSM8909 processor is listed as Widely Supported for an EDL extraction. That makes this phone a perfect candidate for an EDL extraction under Generic options in the UFED. The next determination to make is if the device is encrypted or not. If you are not sure, there is no harm starting with a non-decrypting extraction. It is low level, doesn't require booting, and won't change the hash. That is what I would do for a device like this. Even though this device is running Android 7.1.1, it is cheap and runs an 8GB storage with very low-end specs. In my Mastering EDL Mode guide, I devote a good portion of that paper to determining the likelihood of device encryption. The version of Android is not always a reliable indicator of device encryption. Cheap with very low end specs immediately pushes me toward a non-decrypting extraction. If the device is encrypted, I will find out when I open the extraction and I have the option of using the Generic Qualcomm EDL Decrypting Bootloader.

The last decision to make is how to get the device in EDL Mode. I did find test points on the phone but I first determined I could place the phone in DFU Mode via button combinations. Volume down and volume up while connecting to USB creates DFU Mode. See the short extraction video for the entire process.



CAPABILITY NAME	DESCRIPTION
Lock Pick	Generic lock screen bypass for Android devices running OS version 6 and above with security patch older than August 2018
Mediatek Unencrypted Chipsets	6753, 6735, 6737, 6580, 6795, 6260, 625A, 6592, 6572, 6571, 6752, 6582, 6595, 6261, 6573 & 6583
Qualcomm Decrypting Bootloader EDL	Generic physical extraction, and bypass capability for Encrypted Qualcomm based devices. Generic supported chipsets are: 8909, 8916, 8936, 8939 & 8952. Supported chipset per model: 8917, 8937, 8940, 8953 & 8996
Bootloader EDL	Generic physical extraction, and bypass capability for Qualcomm based devices. Generic supported chipsets are: 8909, 8916, 8936, 8939 & 8952
Bootloader Qualcomm	Physical bypass for Qualcomm based Samsung devices.

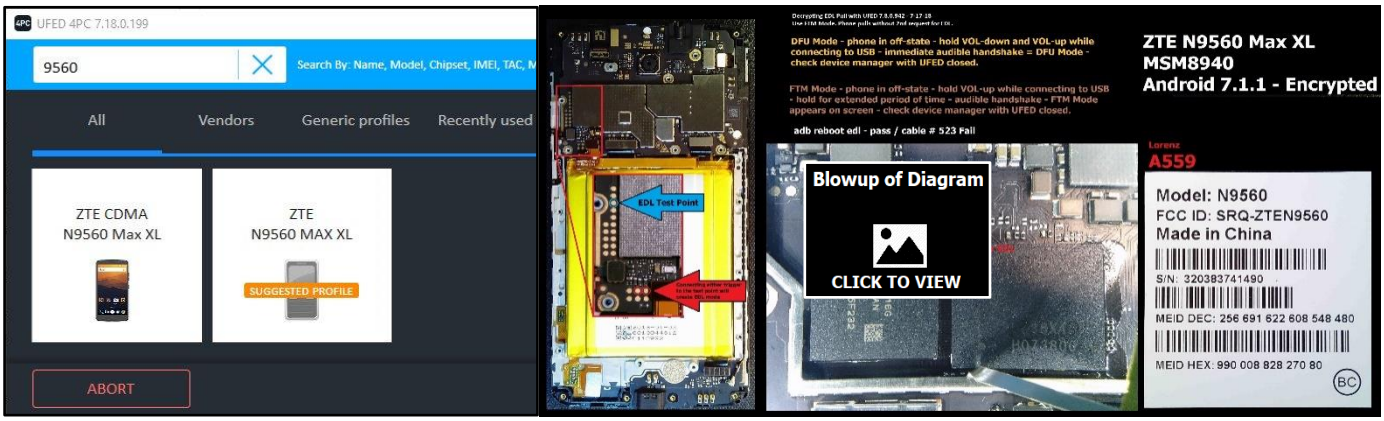
Generic supported chipsets are: 8909, 8916, 8936, 8939 & 8952. Supported

5.6.2 Using Suggested Profiles in the UFED

Suggested profiles in the UFED are an additional feature recently added which serves as kind of a bridge between the library of supported devices in the UFED and the generic options and abilities in the UFED. In many ways, Suggested Profiles are a generic form of extraction but not exactly. Suggested Profiles can appear in a variety of ways when searching for device support in the UFED to include searches for device nicknames, model numbers, and IMEI numbers. Sometimes devices searched will have a supported profile and a suggested profile of the same device next to it. Some searches will yield several Suggested Profiles for a device and it variants. All of this information is helpful in determining strategies of attack on a particular device.

5.6.3 Devices with a specific profile and suggested profile in the UFED

Some devices have a supported profile and a suggested profile in the UFED. The ZTE N9560 Max XL is running a MSM8940 processor. That processor has limited support for EDL extractions in the UFED, meaning some devices running that processor are specifically supported for an EDL extraction. The specific support for the N9560 in the UFED does not include an EDL extraction, only a physical extraction if rooted and no lock bypass option. The Suggested profile in the UFED indicates an EDL extraction may be attempted with no specific instructions for how to get the device into EDL Mode. Lack of specific support in

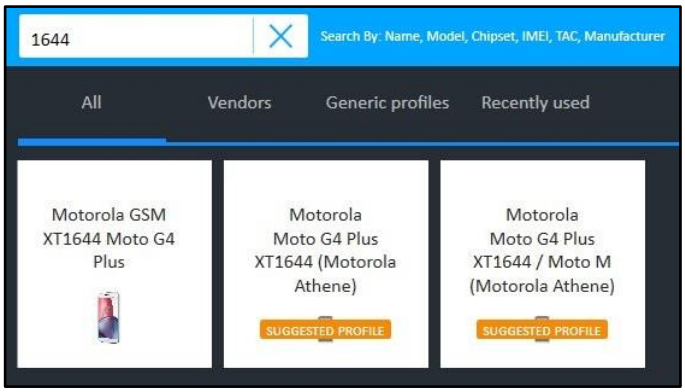


the UFED may be because the Cellebrite didn't have the specific model to test. Lack of support in the UFED for an EDL extraction under a specific device profile may be because there is no way to get the phone into EDL Mode without device disassembly. The solution offered by Cellebrite is to offer a Suggested Profile with general direction and instruction on what type of extraction should be attempted. For the N9560, that specific instruction is that an EDL extraction can be attempted. ***"Note: This method requires the device to successfully enter "Emergency Download" (EDL) mode."*** General instructions in the UFED for a device in this category will include various methods for button combinations and using cable #523.

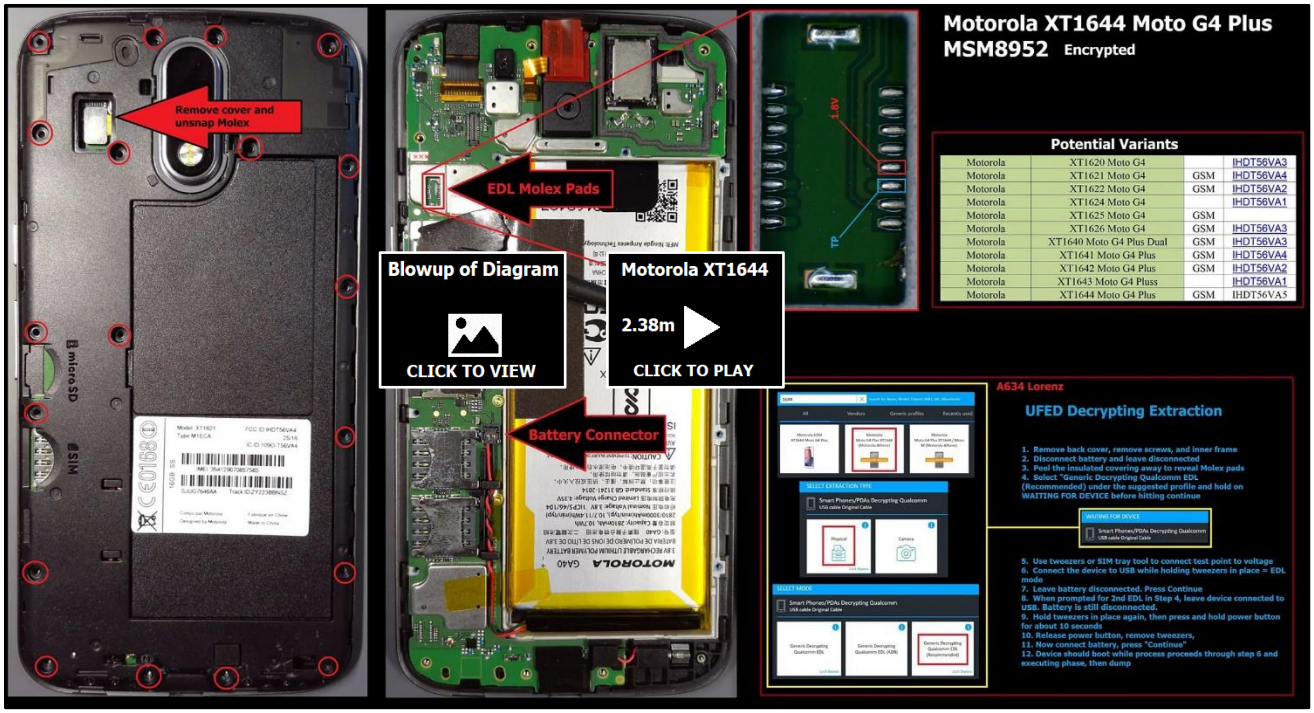
I have tested and extracted this device many times. The device is encrypted by default which means a non-decrypting extraction will not result in readable user data. The UFED suggests a decrypting EDL extraction can be attempted when you click on the Suggested Profile. FTM is the best method of extraction if you're using the UFED. Using FTM mode by holding volume up while connecting to USB, will result in a decrypting extraction and will not require the user to place the device in EDL Mode a second time in Step 4. The N9560 also has a test point that creates EDL Mode but the extraction needs to be a decrypting extraction so any tool used for an EDL extraction will need to have the ability to decrypt.

5.6.4 Using a Suggested Profile and a test point to extract the Motorola XT1644

Using the techniques discussed in this paper, I was able to locate test points on phones in unexpected locations including on Motorola phones. The Motorola XT1644 Moto G4 Plus runs the MSM8952 processor and is encrypted by default. The UFED shows physical lock bypass support under its specific device profile but only for devices with security patches up to May 2017. The device is supported for a physical EDL extraction via ADB under its device profile, but that would require the device to be unlocked with developer options enabled. What if the device is patched beyond the restriction in the bootloader and is not unlocked to enable an ADB extraction? This is another area where Suggested Profiles come into play. There are two Suggested Profiles when searching for the XT1644. Both Suggested Profiles recommend Generic Decrypting Qualcomm EDL extractions.



Before researching test points, placing this phone in EDL Mode would require eMMC faults (shorting) the device into EDL Mode if it was locked. That would require significant disassembly, accessing the underside of the logic board and removal of heat shields. I was able locate the EDL test point on this device on the face-up side of the logic board so all that is required is removing the screws and the back cover. Peeling back a piece of insulation reveals access to the test point I have marked. Once that is located, all that remains is triggering the test point and formulating a strategy for an EDL extraction. By strategy, I mean I need to get the device into EDL Mode in such a way that it allows the UFED to apply the decrypting bootloader and allow the device to boot normally. There is more than one way to place devices in EDL Mode, even with the same EDL method. Whether to start a decrypting extraction with a battery in place can affect the outcome. Sometimes unexpected things can interfere with the creation of EDL Mode. That was the case with this phone in that the battery interfered with the initial creation of EDL Mode using the test point. However, the battery is needed for a successful decrypting extraction, which will require the device to boot normally. It is the case with some phones that a highly specific protocol must be followed for the extraction to succeed. I have described that in the diagram and created a video just for this device.



5.7 Decrypting EDL ability puts the UFED in a class by itself

Nothing has affected the world of mobile forensics as much as encryption over the past few years. Most phones sold today are encrypted by default. There are, of course, a few exceptions to that rule. Extracting devices placed in EDL Mode is a powerful tool but if the device is encrypted, the extraction will result in unusable, encrypted user data if the extraction is a low-level extraction of the device in the off state. The UFED's ability to apply a decrypting extraction to the EDL exploit is one of the UFED's most powerful abilities. It is also what sets the UFED apart from other tools. Of course, the UFED provides many types of decrypting extractions in addition to the EDL exploit but applying a decrypting bootloader to the EDL exploit affects a significant number of devices entering forensic labs today.

One of my favorite options in the UFED is the ability to choose a decrypting or a non-decrypting EDL extraction for Generic EDL extractions. There are still devices sold not encrypted by default today and I receive plenty of older devices likely to not be encrypted. In Mastering EDL Mode, I discussed and listed devices sold with Android 7 and 8 without encryption enabled by default. I discuss tips and clues to determining whether a device is encrypted or not and what strategy to use when determining a decrypting or non-decrypting EDL extraction. There are differences in the two types of extractions in the UFED. A simple distinction between a decrypting vs non-decrypting extraction in the UFED is that a decrypting EDL extraction requires the device to boot normally during the extraction. However, there is much more to consider for a successful decrypting extraction.

5.7.1 Decrypting EDL extractions

Encrypted devices can be exploited just like non-encrypted devices in the UFED. If a non-decrypting EDL extraction succeeds in the UFED under generic options, a decrypting extraction of that same device should be possible also. However, there are times when there will be failures of decrypting extractions of devices that extract via non-decrypting. Decrypting extractions are significantly more complicated behind the scenes with the UFED's decrypting bootloader. For the examiner, a decrypting extraction can require some specific techniques, attention to detail, and at times some creativity. Remember that under generic options, the procedure is not optimized and tested for a specific device. For those who have worked with any significant number of extractions, you know that not all devices behave the same way. There are many things that can interfere with or alter the normal booting of a phone. Variations in devices, damage, booting to charge only mode, and battery levels and functionality are some of the characteristics of phones that can affect a decrypting extraction.

Connection techniques and strategies for placing the device in EDL Mode more than one time during a single extraction are issues the examiner will have to work through on some stubborn devices. I have encountered devices that took hours before I finally found the right combination of strategies and techniques before I was able to get a decrypting extraction to go through. But as tricky as a decrypting extraction can be, I have also been able to perform decrypting extractions on devices thought to be completely destroyed – broken in half, separated from their USB port, damaged battery terminals, and no screen.

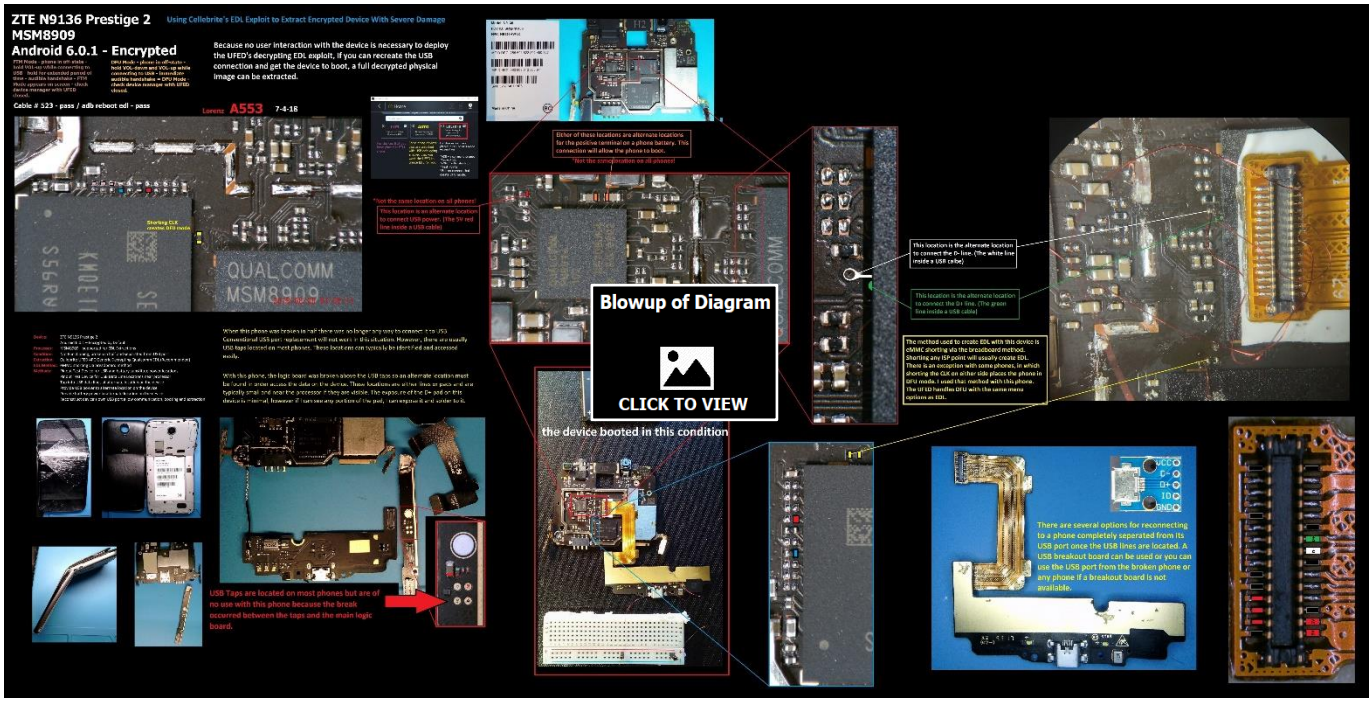
5.7.2 The power of Lock Bypass Decrypting extractions for damaged devices

The most powerful, and often overlooked, aspect of Lock Bypass extractions in the UFED is the ability to extract data from badly damaged, encrypted devices. Before default encryption, JTAG, ISP, or Chip-Off were the go-to solutions for badly damaged devices. With encryption taking those techniques off the table, badly damaged, encrypted devices are challenging. ISP, JTAG, Chip-Off, or any extraction of devices in the off state will yield encrypted user data. There are multiple Lock Bypass Decrypting extractions in the UFED that are what I refer to as hands-free and screen-free extractions. In other words, very little or no interaction is required between the examiner and the device during the extraction process. The UFED's decrypting extraction is in this category. The only requirement from the examiner is that the device be placed in EDL Mode, which can be accomplished

without visual interaction with the phone screen. The only other requirement is that the phone be “capable” of booting normally, but that booting is taken care of by the UFED after exploiting EDL Mode that I create.

If I can get a device, or what’s left of it, connected to the UFED in EDL Mode and in a condition that will allow it to boot, the UFED will take over from there. Whether it is user locked or there is just no way for me to interact with developer options, all I must do is get the phone to live long enough to accept commands from the UFED. There is still significant prep-work and pinning required from the examiner on phones in that condition, which includes alternate USB locations for bypass and power and possibly alternate locations for battery requirements. Most phones can live and boot with no screen. All of that is the easy part of the extraction as the most difficult part takes place with the forensic tool.

I do not take for granted the R & D required to develop such extraction capabilities on encrypted devices. The decrypting EDL extraction falls into this category. Even though there can be issues getting the decrypting bootloader to work on some devices, I have been amazed on what I have been able to do with that tool on some devices written off as a lost cause due to encryption and damage. As an example, see the diagram of a damaged ZTE N9136 in which a decrypting EDL extraction under generic options in the UFED worked. It is also an example of when tools like the USB Finder and the Pinout Jig, discussed early, are invaluable.



5.7.3 UFED GENERIC EDL DECRYPTING GENERAL RULES

1. Works on select devices with Qualcomm processors.
2. Works from EDL
3. Works from DFU
4. Works from FTM
5. Works via ADB on (some) unlocked devices (EDL via ADB can be shut off on some phones)
6. Used on devices that are suspected to be encrypted (Android encryption by default or user encrypted)
7. Works on devices that are not encrypted also with no harm to the device
8. Works on devices that are unlocked, or passcode or pattern locked
9. Will **NOT** work on devices with Secure Startup unless passcode is known
10. Works on devices that are damaged (exception 1 – device must boot)

11. Power and volume buttons not required (exception 1 – DFU/FTM) (exception 2 – power on devices for ADB-EDL) (exception 3 – some devices may require holding power button with user short to create 2nd EDL during some extractions)

5.7.3.1 Decrypting Extraction process

1. EDL / "Continue" / Step 1 / Step 2 / Step 3 / Step 4 / (May require 2nd EDL here) / Step 5 / Step 6 / (booting should start in Step 5 or 6) / Executing Phase / Dump
2. Device must boot normally
3. Special connection or EDL strategies may be required for process to complete or phone to boot
4. The device will extract in the on-state – after booting to the user screen – locked or unlocked
5. No screen is required (exception 1 – ADB-EDL) (exception 2 – some phones require screen to boot) (exception 3 – Secure Startup with known passcode – passcode will need to be entered)
6. Battery is normally required on many devices (exception 1 – rarely devices will boot and extract without battery)

5.7.3.2 Decrypting Failures

1. If device is running processors with "Wide Support" in UFED and fails to extract, check errors below for solution. If no solution, contact Cellebrite tech support as these devices should extract. (MSM8909, MSM8916, MSM8936, MSM8939, MSM8952)
2. Most common – Device failure in Step 1 of 6. Device is not supported for EDL extraction in UFED
3. UFED request to enable USB debugging during connecting phase –
    a. Device was not in EDL Mode when extraction began – will result in Extraction Error – Cannot read phone memory (9)
    b. Device was not supported for EDL extraction
4. UFED fails in connecting phase – device not supported for EDL extraction
5. Failure in Step 6 of 6. Extraction Error – Cannot read phone memory (9)
    a. Device failed to boot, no battery, damage
    b. Phone booted to charge-only mode. A change in connection strategy can remedy this issue many times. Connect phone to UFED in off state with battery connected, wait for charge indicator on device and handshake, leave connected and then create EDL Mode.
6. Failure in Step 1 of 6, after several minutes and then message – Process failed – "We suggest to try another boot loader method."
    a. Examiner failed to remove short before starting extraction.
    b. Device is damaged and is in permanent EDL Mode – close 4PC connect device to a PC with device manager open and see if device is in EDL Mode when you are not shorting it
7. Failure in executing stage.
    a. Contact Cellebrite tech support
8. Failure in dump stage. Incomplete dump or missing data
    a. Contact Cellebrite tech support

5.7.4 Non-decrypting EDL extractions

For devices which are not encrypted, low-level (non-booting) extractions of devices in the off-state is my preferred, first-strike exploit, especially if I receive the device in the off-state under unknown circumstances. Damaged phones or phones that have been in storage for an extended period of time can have unseen issues in which charging or booting could further damage the device. Therefore, I may choose to disassemble a phone to reach a test point or to create an eMMC fault, as opposed to other available methods that do not require

disassembly, but require charging and booting the device. A non-decrypting EDL extraction may require a screwdriver but a bootloader requires a working phone to boot normally. So which method has the least risk? That depends on the level of disassembly needed to create EDL Mode and examiner experience.

Both are forensically acceptable forms of extraction but have different considerations. A non-decrypting EDL extraction in the UFED is one of the easiest extractions to perform with most devices. It is low-level and most of the time will not require a battery in the device. All of that means you can extract the device in the exact condition in which it was received – in the off state. The most challenging part of the extraction is knowing what method is required to create EDL Mode, if any disassembly is required, and how much disassembly. These considerations were some of the reasons I went looking for test points on devices I had already extracted via eMMC faults. I was searching for an easier path, requiring less disassembly. I included this list of devices sold with Android 6.x or higher, but without default encryption, in my Mastering EDL Mode guide last year. None of the diagrams I made last year included any information on test points. I have now changed 10 of them to include the test point locations. Of course, there are hundreds of devices still in use today that are not encrypted and can be exploited by an EDL extraction. Most of those devices will have test points.

Devices Sold With Android 6 or Higher That Were Not Encrypted by Default						
Phone	**Processor**	**OS Version**	**RAM**	**Storage**	**Carrier**	**Specs**
Alcatel 5044R Ideal Xcite	MSM8909	7.0	2GB	16GB	ATT Prepaid	PS
Alcatel A577VL	MSM8909	7.1.1	2GB	Up to 16GB	Straight Talk	PS
ATT Axia QS5509A	MSM8909	8.1.0 (Go)	1GB	16GB	ATT Prepaid	Best Buy
Coolpad Illumina 3310A	MSM8909	8.1.0 (Go)	1GB	8GB	Sprint	PS
Kyocera Cadence S2720PP	MSM8909	7.1.1	2GB	16GB	Verizon	PS
Motorola XT1609 Moto G4 Play	MSM8916	6.0.1	2GB	16GB	Verizon	PS
ZTE N9137 Tempo X	MSM8909	7.1.1	1GB	8GB	Virgin Mobile	PS
ZTE Z719DL ZMAX One	MSM8909	7.1.1	2GB	16GB	Tracfone	Z719DL
ZTE Z835 Maven 3	MSM8909	7.1.1	1GB	8GB	ATT Prepaid	PS
ZTE Z839 Blade Vantage	MSM8909	7.1.1	2GB	16GB	Verizon	PS
ZTE Z852 Fanfare 3	MSM8909	7.1.1	1GB	8GB	Cricket	PS
ZTE Z899VL Majesty Pro Plus	MSM8909	7.1.1	2GB	16GB	Straight Talk	PS
ZTE Z963VL Max Duo	MSM8952	6.0	2GB	16GB	Tracfone	PS

5.7.5 UFED GENERIC EDL NON-DECRYPTING

1) Works on select devices with Qualcomm processors.
2) Works from EDL
3) Works from DFU
4) Works from FTM
5) Works via ADB on (some) unlocked devices (EDL via ADB can be shut off on some phones)
6) Used on devices that are suspected to not be encrypted
7) Works on devices that are encrypted (No harm but user data will be encrypted)
8) Works on devices with Secure Start (No harm but user data will be encrypted)
9) Works on devices that are unlocked or passcode or pattern locked
10) Works on devices that are damaged or function normally
11) Power and volume buttons not required (exception 1 – DFU/FTM) (exception 2 – power on devices for ADB-EDL)
12) Is low-level and will not change the hash (exception 1 – ADB-EDL)

5.7.5.1 Non-decrypting Extraction process

1. EDL / "Continue" / Step 1 / Step 2 / Dump
2. No booting will occur (exception 1 – ADB-EDL)
3. The device will extract in the off-state (exception 1 – ADB-EDL – start from on state)
4. No screen is not required (exception 1 – ADB-EDL)
5. No battery is required on many devices (exception 1 – ADB-EDL) (some devices need battery for test points)
6. The hash of the image will not change (exception 1 – ADB-EDL)

5.7.5.2 Non-decrypting Failures

1) If device is running processors with "Wide Support" in UFED and fails to extract, check errors below for solution. If no solution, contact Cellebrite tech support as these devices should extract. (MSM8909, MSM8916, MSM8936, MSM8939, MSM8952)
2) Most common. Device failure in Step 1 of 2. Device is not supported for EDL extraction in UFED
3) UFED request to enable USB debugging during connecting phase – No EDL support for device, can also happen if device was not in EDL Mode when extraction started
4) Failure in Step 2 after several minutes and then message – Process failed – "We suggest to try another boot loader method."
   a) Examiner failed to remove the short before an extraction
   b) Device is damaged and is in permanent EDL Mode – close 4PC connect device to a PC with device manager open and see if device is in EDL Mode when you are not shorting it
5) Failure in dump stage. Incomplete dump or missing data
   a) Contact Cellebrite tech support

5.7.6 Compare non-decrypting and decrypting extractions on the same device

It is helpful to see the difference between non-decrypting and decrypting extractions by performing both on the same device. There is also a need to do this with some devices. Examiners will receive some devices not believed to be encrypted or not known and thus perform a non-decrypting extraction only to find the device is encrypted. The non-decrypting extraction is harmless in this situation as it will not damage an encrypted device and it will not change the hash. It will then be necessary to perform a decrypting extraction on the same device. That means a battery will be needed for the extraction and the device must boot normally. The Z799VL Majesty Pro is encrypted by default and the Z899VL Majesty Pro Plus is sold not encrypted. These two devices share the same FCCID and have identical logic boards, but they require completely different strategies for extraction.

Phone	Processor	Encrypted	Test Point Used	Video of Extraction in UFED Generic Options
ZTE Z899VL	MSM8909	No	Yes	Non-decrypting EDL
ZTE Z799VL	MSM8909	By default	Yes	Decrypting EDL
Alcatel A574BL	MSM8909	No	Yes	Non-decrypting EDL
Alcatel A574BL	MSM8909	User	Yes	Decrypting EDL
ZTE Z852	MSM8909	No	Yes	Non-decrypting EDL
ZTE Z852	MSM8909	User	Yes	Decrypting EDL

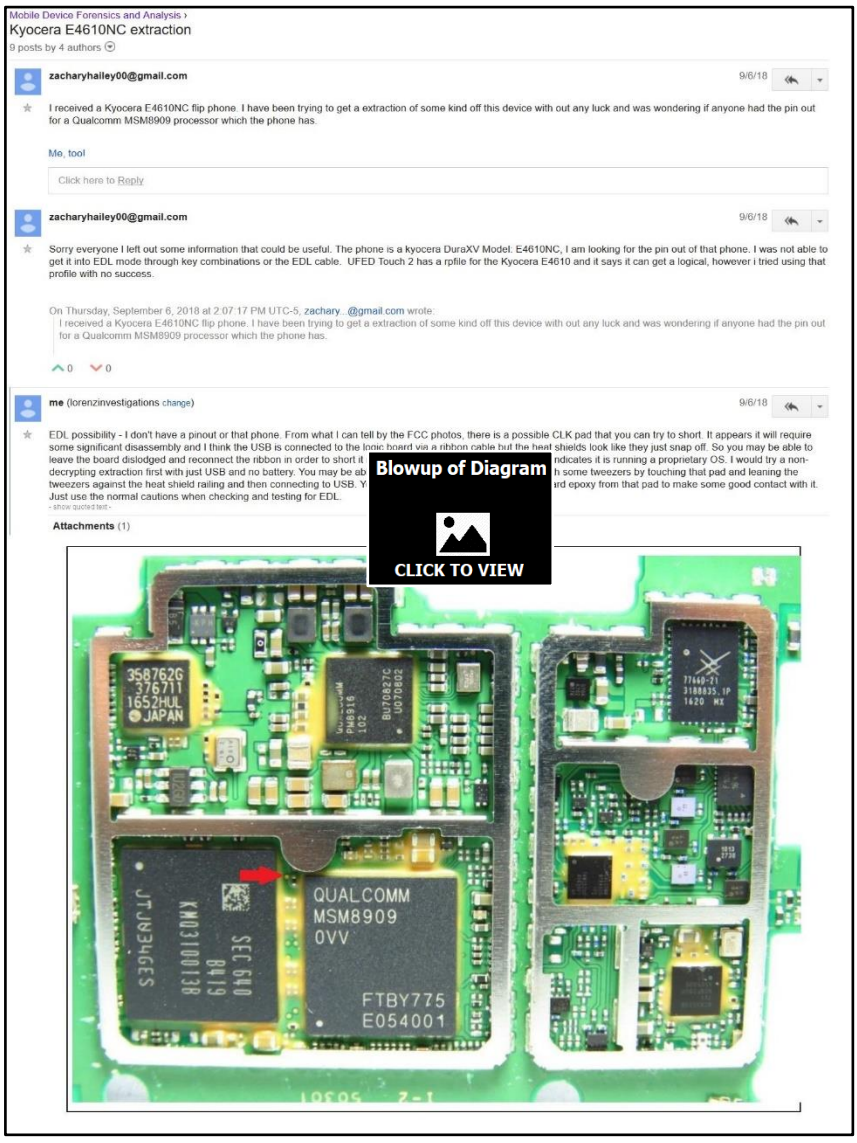
6 Bringing everything together – Kyocera E4610 case study

Sometimes the best way to understand all of the processes and techniques involved is to examine the research and extraction of a particular device from start to finish. The devices we all encounter are different and have different issues and possibilities, but the process of intake, identification, research, pinning, cooperation, testing, and sharing is the same with all devices. Sharing and asking questions on forensics forums helps everyone learn and test available tools and skills. The sharing of information about successes and failures benefits everyone.

6.1 Initial question posted on MDFA

In September of 2018, a member of MDFA posted a question regarding extraction possibilities for the Kyocera E4610NC – DuraXV. He checked Cellebrite's UFED to find only logical support for the device but he did further research and discovered the device is running a MSM8909 processor. That processor is one Cellebrite list as "Widely Supported" for EDL extractions.

That means a lock bypass physical extraction is possible with this device using the UFED's Generic options. At the time of the post, I did not have that device to test but I did pull up a photo from the FCC website and marked what I thought was a possible location to create and an eMMC fault or "short" the device into EDL mode. From examination of only the FCC photos and previous work with other similar models, I thought this would be somewhat of a challenging extraction due to teardown requirements and the presence of epoxy. In 2019, other examiners posted success with an ADB EDL extraction but questions remained about shorting for locked or damaged devices.



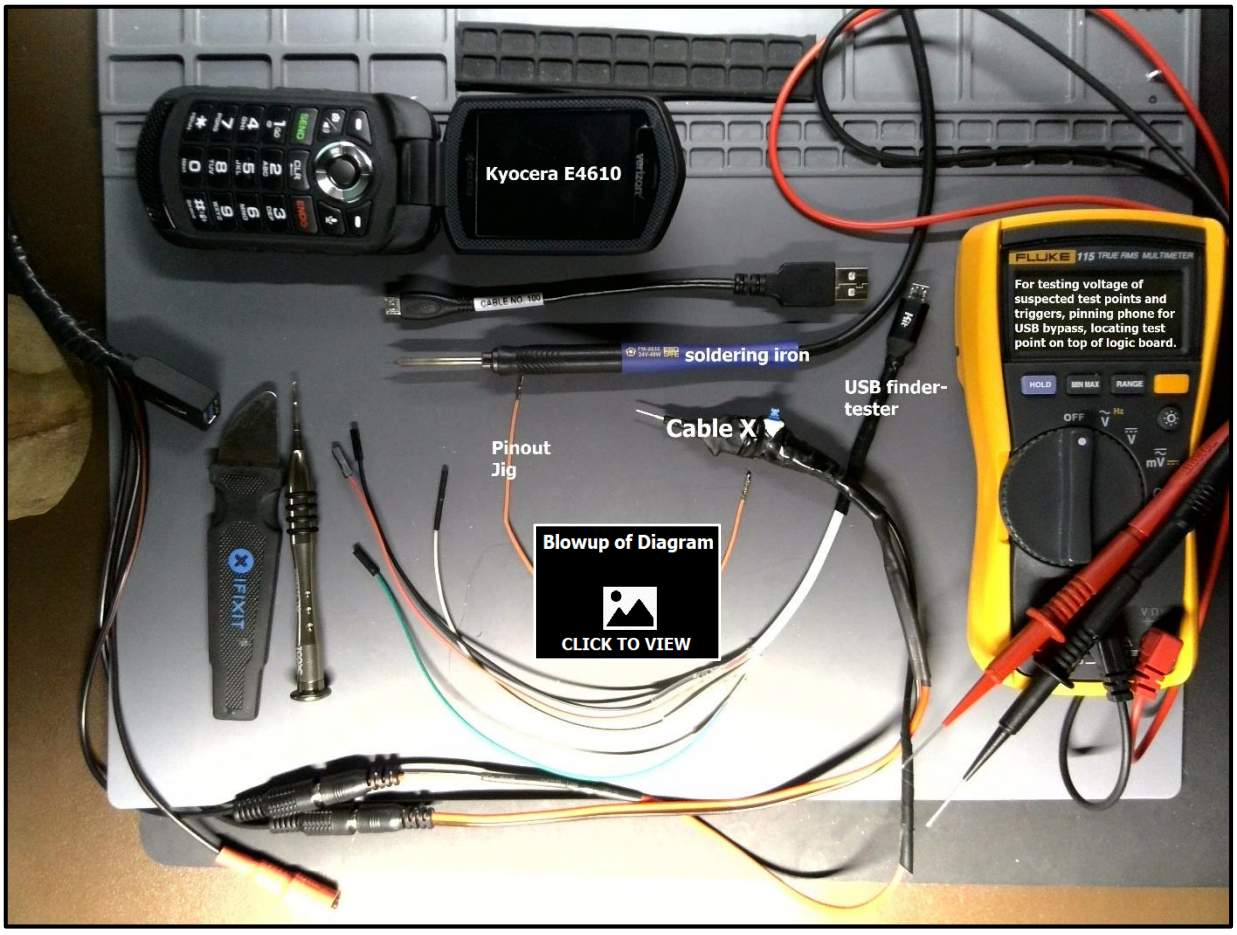
6.2 Identification and evaluation of Kyocera E4610

In May 2019, I obtained a Kyocera E4610 for testing. My purpose was to research and locate a possible test point as part of my test point research. I tested the suspected location I marked on the FCC diagram in my 2018 post and confirmed that shorting that location creates EDL Mode. However, I also wanted to find a test point that would eliminate the need to remove the logic board and reduce the amount of disassembly required if possible.

6.3 Testing the Kyocera E4610 for least invasive methods of creating EDL Mode

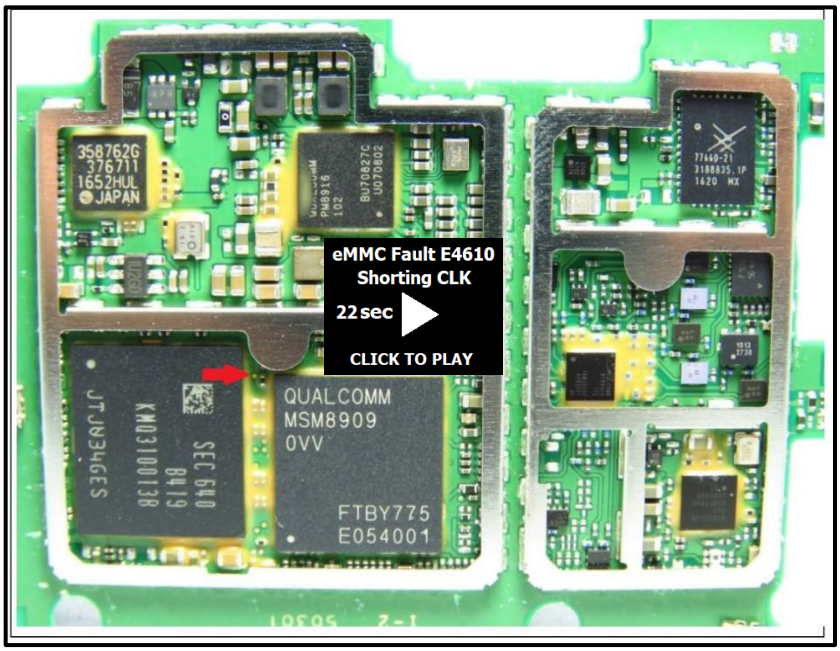
Before tearing into an unknown phone looking for test points I test for other methods of creating EDL Mode to include button combinations, ADB, and use of the EDL cable – Cellebrite's cable #523. Of course, finding that one or more of those methods work will not stop me from researching further on the same phone – provided the phone is a test phone and not evidence. The reason I research more than one method to create EDL Mode is because some methods may not work on every phone of that same model or variants of that device. Damage can prevent button combinations from working and EDL via ADB usually requires an unlocked and functioning device. Therefore, I don't assume that one method will always work for every phone. Cable #523 did not produce EDL Mode on this E4610. Button combinations also failed to create EDL, DFU, or FTM.

When researching a test phone I assume a real world worst case scenario by asking "what if…". That means I search for solutions that will allow me into a locked phone, a damaged phone, a phone that is locked with no screen or USB port, etc. All of that means, when I have time, I will diagram whatever I can on a test phone to include USB bypass locations and multiple methods for creating EDL Mode. I may not need it today, but I or someone else will need it.



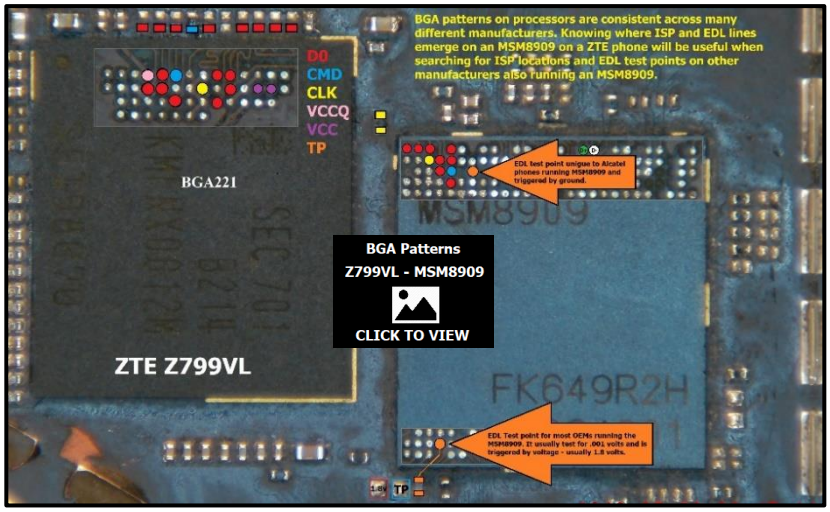
6.4 Testing my guess for eMMC short on the Kyocera E4610

I made a guess about the location of a short point based on looking at an FCC photo of this device in 2018. I wanted to see if my guess was correct. It was also a chance to use Cable X and demonstrate why I picked that location. I disassembled the device and tested the point I marked on the FCC diagram. I suspected this location was the CLK because of its location in relation to the processor and because it was a resistor with a pad located next to it.



6.5 Using the ZTE Z799VL to exploit the Kyocera E4610

Looking at a diagram of the ZTZ Z799VL next to the FCC photo of the Kyocera E4610 and it is easy to become less impressed with the guess I made just by looking at a photo. It wasn't much of a guess. Both phones run the same MSM8909 processor and both run BGA221 eMMC storage. For more on why and the extent to which these patterns exist, review a previous relevant section (*2.3.4 Is the processor built for the phone, or the phone built for the processor?*).



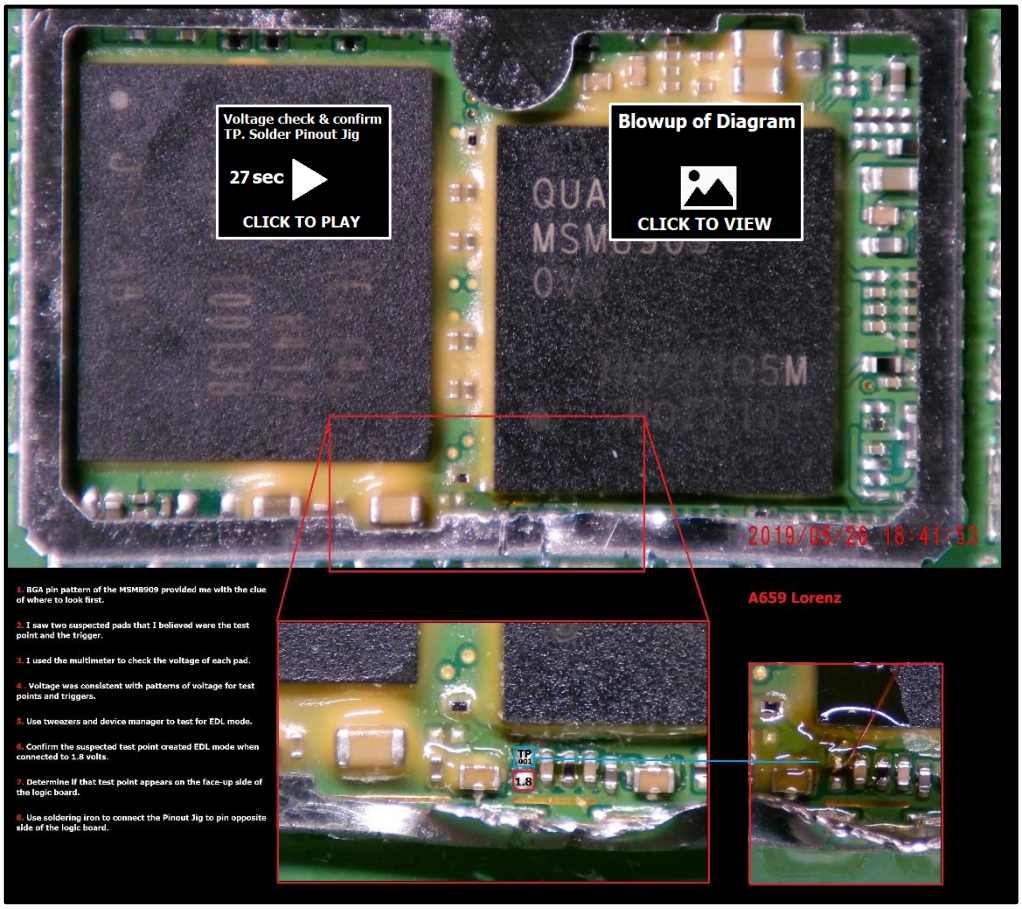
There are patterns just like this all throughout the universe of phones. Some are not so obvious but the more you research and the more you look at logic boards, the more you know where to look when you don't have a pinout handy. Knowing the BGA of the MSM8909 and therefore the location of ISP pins under the processor is useful for locating ISP locations without chipping the storage and when no pinout is available for the device. My x-ray view of the Z799VL is like staring through the top of both chips. Because the point I marked on the E4610 is an ISP point (CLK) and not a test point, I know that applying ground to the CLK, CMD, or Data points is the most reliable and safest way to create EDL Mode (absent a known test point). The CLK was the most accessible point I saw in the FCC diagram and therefore was my first recommendation for an attack. So when finally got the E4610 in my hands, I applied ground to the marked pad on the E4610 with the probe on Cable X set to ground, it immediately created EDL Mode. The takeaway from this is that you don't have to confine yourself to a specific pinout of the exact model phone you receive as evidence, in order to make an educated guess about what point will create EDL Mode.

6.6 Finding the test point on the Kyocera E4610

Processor patterns don't just apply to ISP locations. These patterns are found with test points also. Using the pinout from the Z799VL with the BGA pin pattern from the MSM8909 gave me the exact location of the CLK on the Kyocera E4610. That doesn't mean the pattern regarding the test points on devices running the MSM8909 will be as consistent, but "the known" is a good place to start looking. Remember that I could easily chip the processor from the E4610 and use my MSM8909 pinout to pin (trace) to the test point on the phone and be done with it. However, I always try to locate the test point without chipping, just as I do with ISP. This way I can test what I found if I only have the one phone to test. This is where all of the steps, methods, and tools discussed in this paper come in handy. This is also another chance for me to extol the benefits of ISP, JTAG, and Chip-Off training and practice.

The bottom of the MSM8909 is where the BGA test point pin is located. One row up and four pins from the left side of the MSM8909. The test point is located in a similar location on the MSM8916. Although this BGA location on the MSM8909 can lead to the test point on Alcatel phones that is triggered by voltage, the more common test point on Alcatel phones that is triggered by ground is found at the top of the MSM8909. That Alcatel test point may lead to specific locations on other OEM's but I have yet to be able to get it to create EDL Mode by any means. Since this is a Kyocera phone, it plays by the same rules as all other phones. It should have one test point, triggered by voltage. When I looked at this under the microscope I could immediately see two pads that I suspected were the test point and the trigger. After bending back the heat shield rail, I was able to confirm the voltage. Having one test for .001v and the other testing for 1.8v, made me confident in my choice. Remember that it is not just the voltage by itself, but the location also.

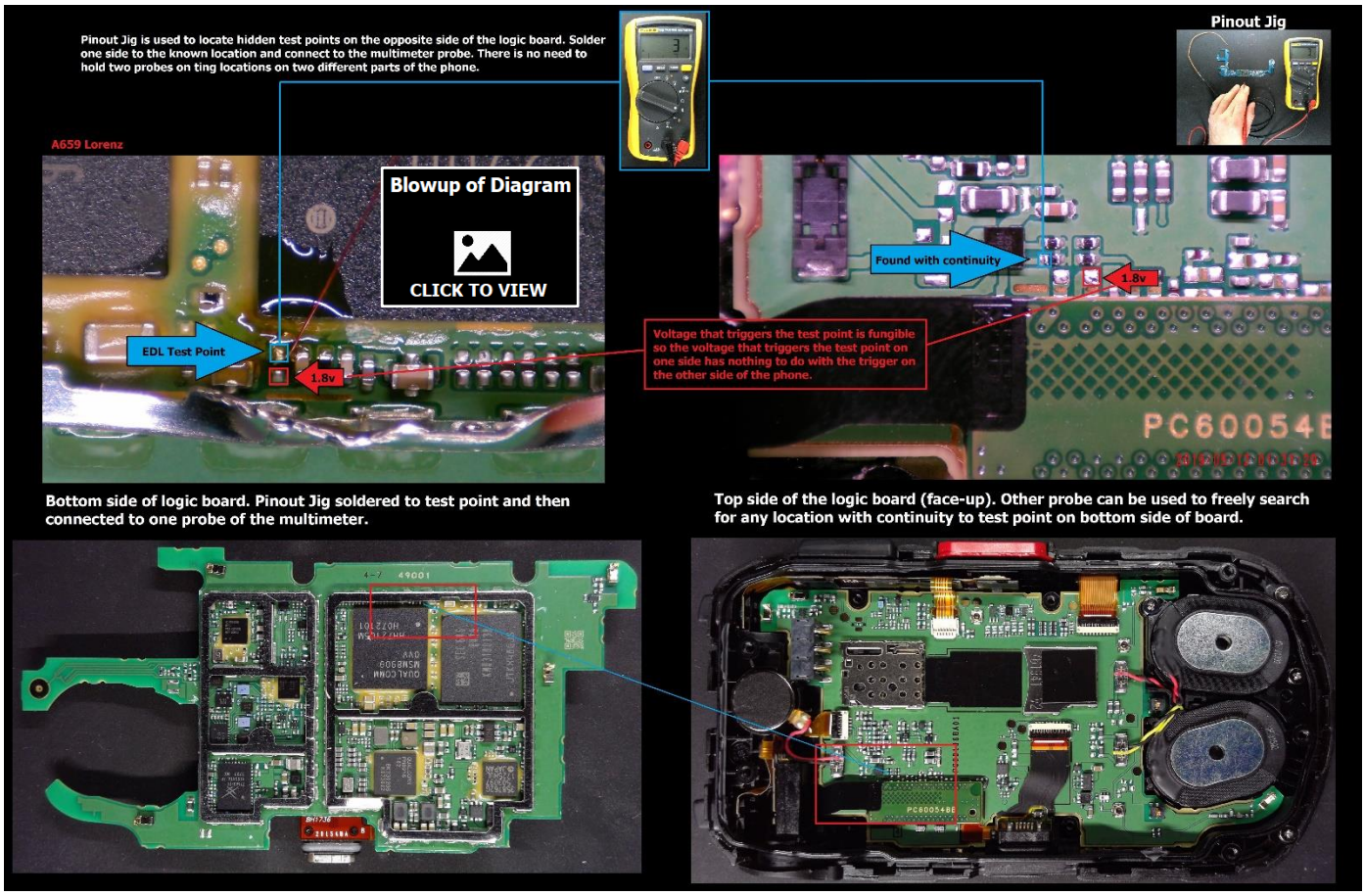
After seeing the voltage on both pads, I gave it the final test, which was to connect the trigger to the test point and then plug into USB. When I did, EDL Mode was created immediately. But how does this test point help me? It's on the same side as the CLK pad that has already proven to create EDL Mode for me. The test point was actually more difficult to access because of the heat shield railing. If this test point is to be truly useful, it should put the examiner in a better place logistically. That is where another homemade tool comes into play.



6.7 Finding an alternate location for the same test point on the Kyocera E4610

There is only one test point, as I have defined it in this paper. Of course, Qualcomm or phone manufacturers are welcome to come out and disagree with me and share some information if they like. Alcatel seems to be the exception of having two test points, one shared by all other manufacturers, triggered by voltage, and one triggered by ground, unique to Alcatel. So once I located the test point on the E4610 just below the processor, I was not looking for another test point. I was looking for the same test point emerging in another location. If test points are designed to be convenient, at least more convenient than eMMC faults, then we can assume there will be two perfectly round pads somewhere on the face-up side of the logic board waiting for a pair of tweezers. But that is not always the case. Sometimes the test point I find looks nothing like a test point. In fact, the locations I find were probably not meant to recognized or used, it just happens to be a location where the line come out of the board.

The test point I located near the processor on the E4610, is certainly not in an easily accessible location and certainly not easy to find unless you know something about what and where to look. However, the test point and the trigger are both pads located next to each other, which seems to be intentional. That means what I found below the processor was the intended test point and trigger. So what I am either looking for something that is better (round pads) or the other alternative, an accident.

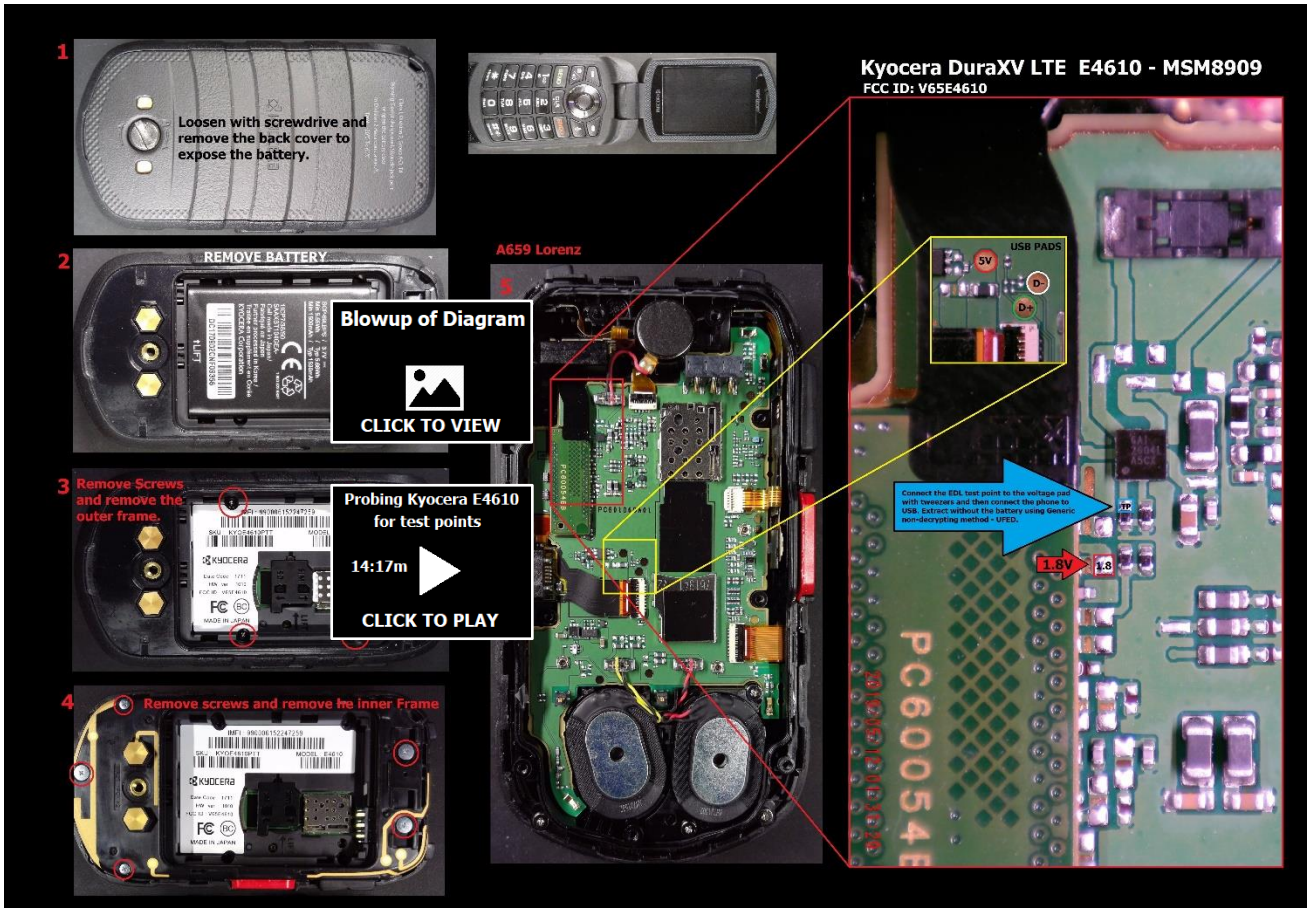


On other phones where I located the first emergence of the test point at the bottom of the MSM8909, I was able to find the final destination elsewhere. It was not always easy as the line passed through some resistors on some phones, so pinning was challenging. The locations involved here are tiny and it would be nearly impossible to hold one probe of a multimeter on one side of the logic board while using the other probe to search the other side. The

pinning jig is something I have been using for years to pinout processors and phones where not everything I was looking for was close by my known location. The tool is very simple but something I use almost daily. In the previous section, I soldered that jig to the test point just below the processor on the E4610. Now I just need to connect the jig to one probe on my meter (either one doesn't matter for continuity) and use the other probe to freely search the opposite (face-up) side of the logic board. In the video associated with this phone, I also demonstrate the use of the USB Finder. The USB Finder is another simple tool that makes it easy to verify you have a function port and to help troubleshoot and trace USB lines.

6.8 The final result has made the Kyocera E4610 easier and safer to extract

I expected to find the test point on this phone and it was exactly where I thought it would be. It was my first guess on the underside of the logic board. I suspected it may be available on the face-up side of the logic board but was not sure it would be and had no idea where it would be. This phone is but one of many examples of why many test points are unknown and have gone unnoticed. All of the tools, techniques, and procedures used to locate and mark the EDL test point on the E4610 will work on many different phones. An extraction of this phone will be easier for me the next time I encounter it and similar models. Each time I do research on one phone I learn something that helps me get into another phone. Locating test points and finding methods to get extractions to work is different for every phone. Some are much easier than this phone and some are much more difficult. The processes involved and tools used will benefit examiners in many ways and on many different extractions. The effort for me has always been worthwhile, even if I fail.



7 EDL characteristics, patterns, and pinouts for specific OEMs

The following sections will cover specific OEMs and will include diagrams and pinouts of devices organized by particular manufacturers. I have only included devices from my database for which I have identified test points. Please keep in mind that this is not an all-inclusive database. In fact, it only scratches the surface regarding devices with test points. I am limited to phones to which I have access or that come across my path in course of doing forensics. I have made my own color code based on support and exploits in Cellebrite's UFED, but that constantly changes and it is not the only tool I own.

7.1.1 Reminder About Device Variants

When I pinout a device, it is normally based on a device I have in my inventory or I have personally examined. I will generally label the diagram with the specific model number of the device. With variants I believe have the same pinout, I use the original diagram I created with the original name I assigned to the pinout. That way I know the pinout I am using for a particular variant was actually created by pinning a different model from the same family of phones. The screenshot from my database tells me that I pinned the Motorola XT1609 Moto G4 Play, but I believe it will also work for the other variants listed.

EDL Wide Support	8909 8916 8936 8939 8952	Samsung Decrypting Qualcomm MSM - 8917 8937 8952 8953 8976					See Processors Tab for details on exploits		
EDL Limit Support	8917 8996 8937 8940 8953	Samsung Legacy							
MTK Decrypting	6580 6737 6750 6753 6755 6757 6797 6735	HiSillicon Kirin Decrypting - 620A 64bit, 650-568, 659, 93X, 95X, 96X, 620, 910, 920, 925, 930, 935							adb shell cat /proc/cpuinfo
Vendor	Model	GSM CDMA	FCC	FCC Photo	FCC Name	Presumed Processor	Date Added UFED	Lorenz Inventory	Lorenz ISP Pins
Motorola	XT1601 Moto G4 Play		IHDT56VD1			MSM8916			ISP-XT1609
Motorola	XT1602 Moto G4 Play		IHDT56VD3			MSM8916			ISP-XT1609
Motorola	XT1604 Moto G4 Play		IHDT56VD5			MSM8916			ISP-XT1609
Motorola	XT1607 Moto G4 Play	CDMA	IHDT56VD6			MSM8916	11/2/2016		ISP-XT1609
Motorola	XT1609 Moto G4 Play	CDMA	IHDT56VD4			MSM8916	9/5/2017	A499	ISP-XT1609

I think I am correct when I create a pinout and list variants but I can be wrong. It is not always easy to determine what processor a particular device is running. Some variants run more than one model of Qualcomm processor. So be prepared to do your own research and validation. Phones can vary depending on where in the world they are released and I have seen examples where two identical models behave differently in terms of extraction support based on where in the world they were purchased.

EDL Wide Support	8909 8916 8936 8939 8952	Samsung Decrypting Qualcomm MSM - 8917 8937 8952 8953 8976					See Processors Tab for details on exploits		
EDL Limit Support	8917 8996 8937 8940 8953	Samsung Legacy							
MTK Decrypting	6580 6737 6750 6753 6755 6757 6797 6735	HiSillicon Kirin Decrypting - 620A 64bit, 650-568, 659, 93X, 95X, 96X, 620, 910, 920, 925, 930, 935							adb shell cat /proc/cpuinfo
Vendor	Model	GSM CDMA	FCC	FCC Photo	FCC Name	Presumed Processor	Date Added UFED	Lorenz Inventory	Lorenz ISP Pins
Motorola	XT1760 Moto E4		IHDT56WC7	FCC		MT6737			
Motorola	XT1761 Moto E4		IHDT56WC5	FCC		MT6737			
Motorola	XT1761 Moto E4	GSM					2/7/2019		
Motorola	XT1762 Moto E4	GSM	IHDT56WC6	FCC		MT6737	3/21/2018		
Motorola	XT1763 Moto E4 (Brazil)		IHDT56WC4	FCC		MT6737			
Motorola	XT1765 Moto E4		IHDT56WC2	FCC		MSM8917		A565	ISP-XT176x
Motorola	XT1766 Moto E4 (Europe)					MT6737			
motorola	XT1766 Moto E4 (USA)	GSM	IHDT56WC3	FCC		MSM8920	5/1/2018	A609	ISP-XT176x
Motorola	XT1767 Moto E4	CDMA	IHDT56WC1			MSM8917	4/18/2017	A418	ISP-XT176x
Motorola	XT1768 Moto E4		IHDT56WC1	FCC		MSM8917			ISP-XT176x
Motorola	XT1769 Moto E4		IHDT56WC8	FCC		MT6737			

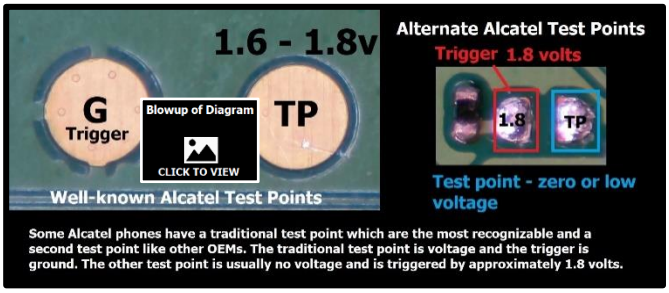
7.2 Alcatel

The most commonly known Alcatel test points are voltage and the trigger is ground. Some Alcatel phones have two separate test points: one, which responds to ground, and one, which responds to voltage.

Commonly Know Alcatel Test Point

Test Point – either no volts or 1.6 – 1.8 volts

Trigger – either ground or voltage

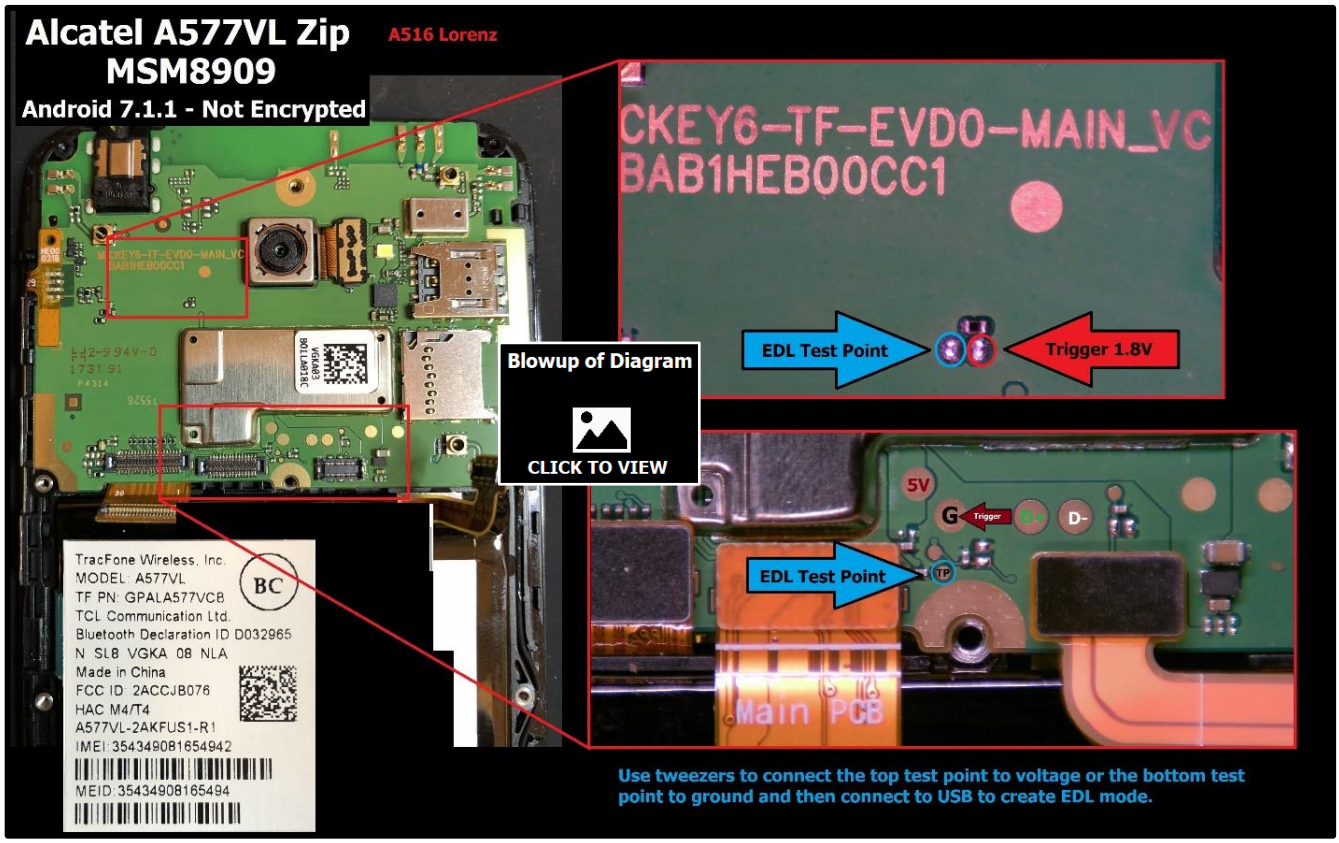


Addition Alcatel Test Point (consistent with other OEMs)

Test Point – no (low) voltage

Trigger – 1.8 volts

Alcatel phones will usually attempt to boot without a battery. They will usually start to boot then power down and start again while connected to USB. Locations can be tested for voltage while the phone cycles. Applying tweezers to the trigger and test point during these cycles will usually trigger EDL Mode when the phone cycles and attempts to boot. Several Alcatel models have both test points easily accessible with the removal of the back cover. The test point which is triggered by voltage leads to a pin under the processor shared by LG and many other models running that same processor. The test point triggered by ground leads to a pin under Qualcomm processors that appears to trigger EDL only on Alcatel phones. I have not be able to identify any other OEMs with a test point triggered by ground but that is only based select devices I have tested.

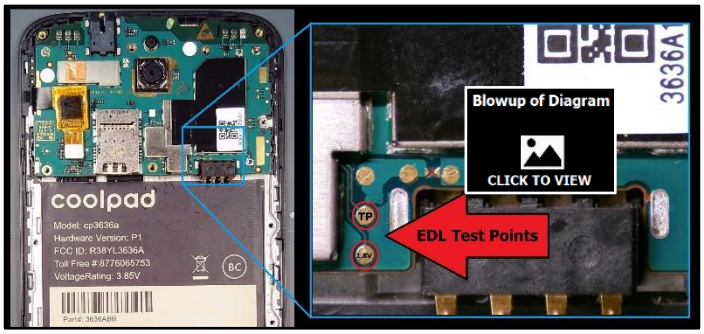


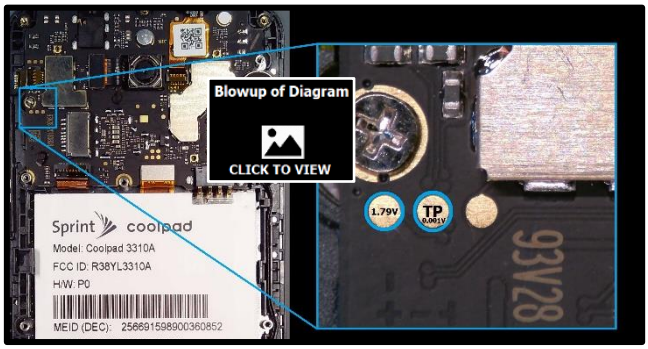
7.3 Coolpad

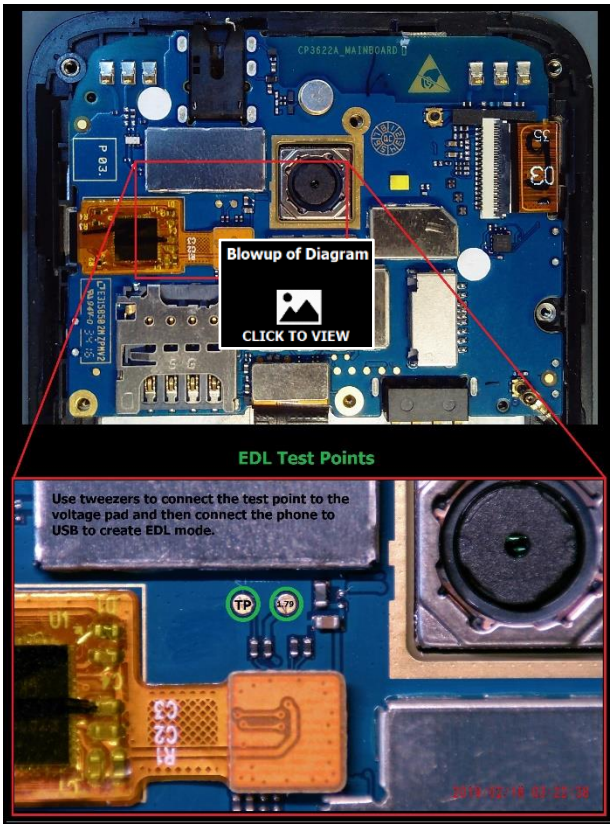
Coolpad devices behave like all other OEMs with a test point which is no or low-voltage triggered by 1.8 volts, usually located next to the test point. Test points on devices I tested were easy to access and disassembly of Coolpad devices was relatively easy, only requiring screw removal. Other options for creating EDL like EDL cables and button combos should be explored before disassembly and looking for test points.

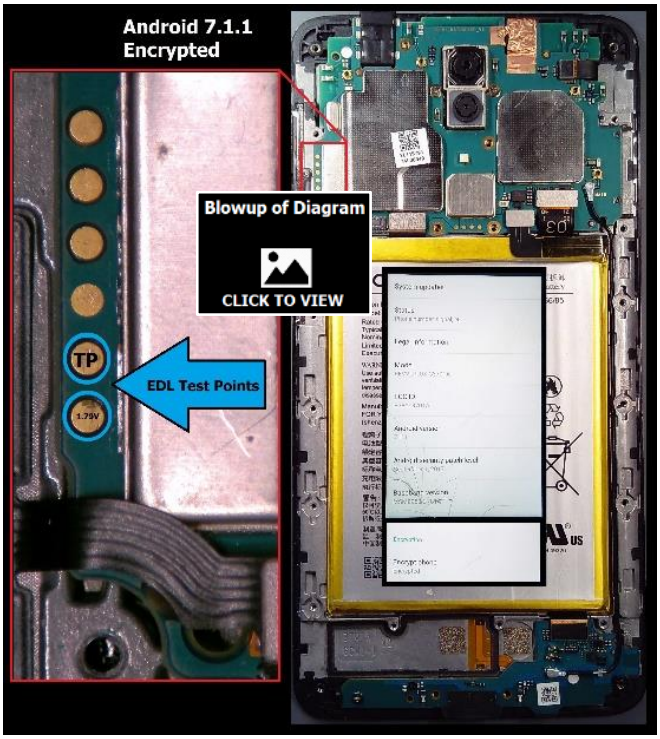
Test Point – no voltage

Trigger – 1.8 volts









7.4 Google Pixel

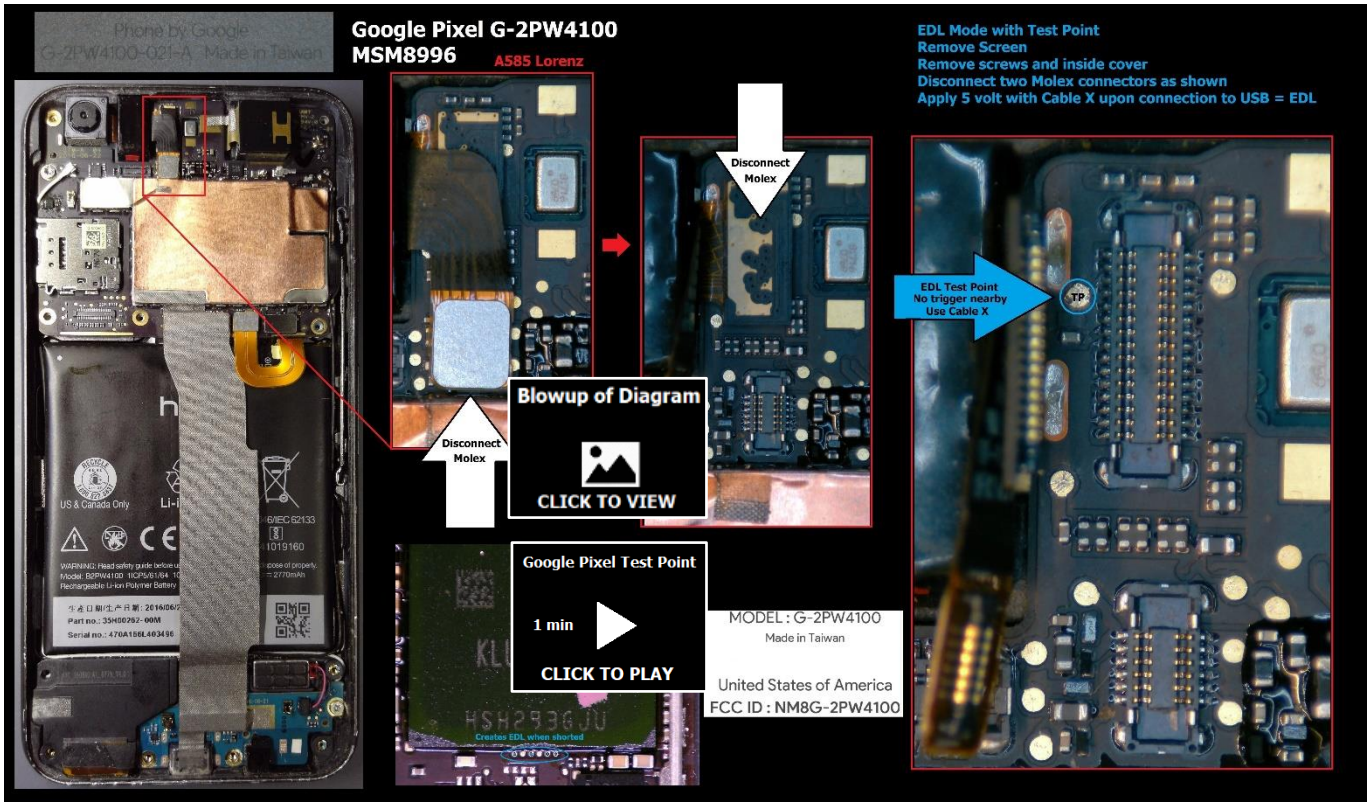
The Google Pixel is running the MSM8996 and this device is not supported for an EDL extraction at this time. It does have an EDL test point I verified after some experimentation. I located the test point by chipping an MSM8996 on a broken Pixel and reverse pinned from a known test point on that processor. I marked the test point location under the MSM8996 by chipping and pinning from a known test point on an LG LS992.

There are several other pads in the vicinity of the test point on the Pixel and some of those pads tested for over 2 volts. None of them, to include pads test for 1.8 volts, would trigger EDL Mode when applied to the test point. It wasn't until I applied voltage using Cable X, that I was able to trigger EDL Mode. I can do so reliably and repeatedly with Cable X, which is applying 5 volts.

Should there be a need to use this test or a Firehose loader for this device in the future, the test point can be accessed without removing the logic board. The device can have the battery connected and disconnected from the position seen in my pinout so triggering the test point and boot should be easy from this position. The most significant issue with accessing this test point is removing the screen on the Pixel. Screen removal is easy but requires some expertise and care not to damage it. The Pixel screen is very durable, but only while installed. It is very fragile when removed.

Test Point – no voltage

Trigger – over 2 volts. 5 volts from Cable X works consistently

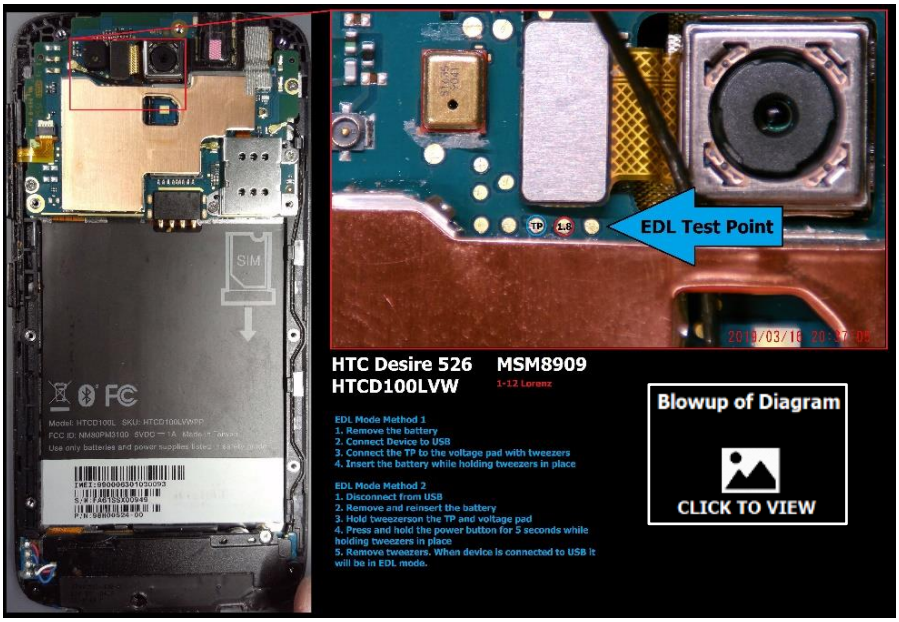


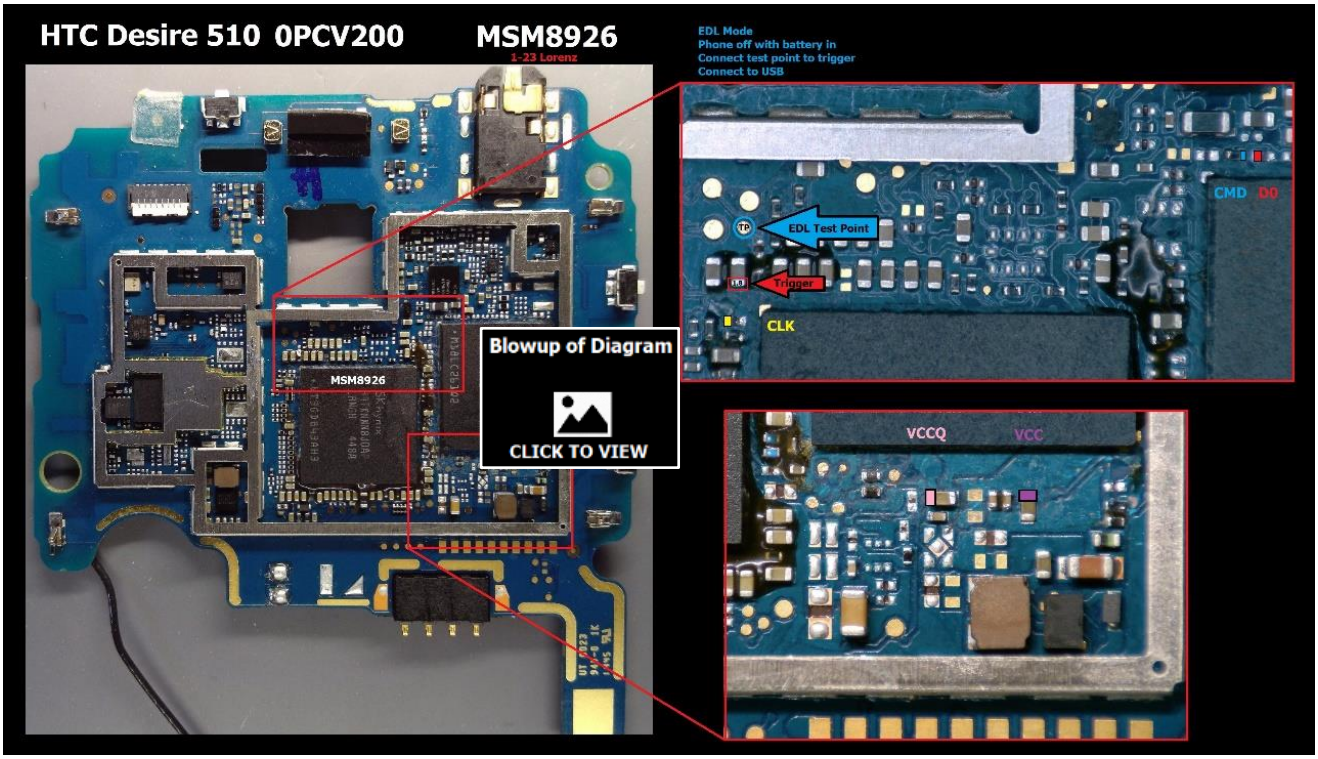
7.5 HTC

I had limited access to HTC devices but did locate test points connected to the same BGA pin as other OEMs and with the same voltage and trigger requirements.

Test Point – no/low voltage

Trigger – 1.8 volts



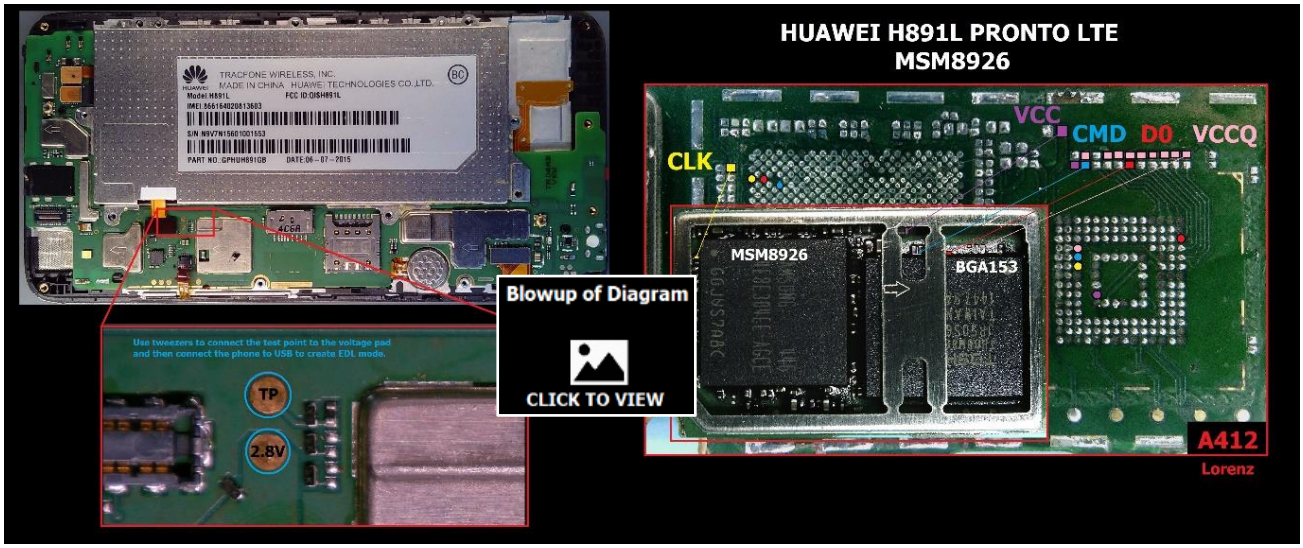


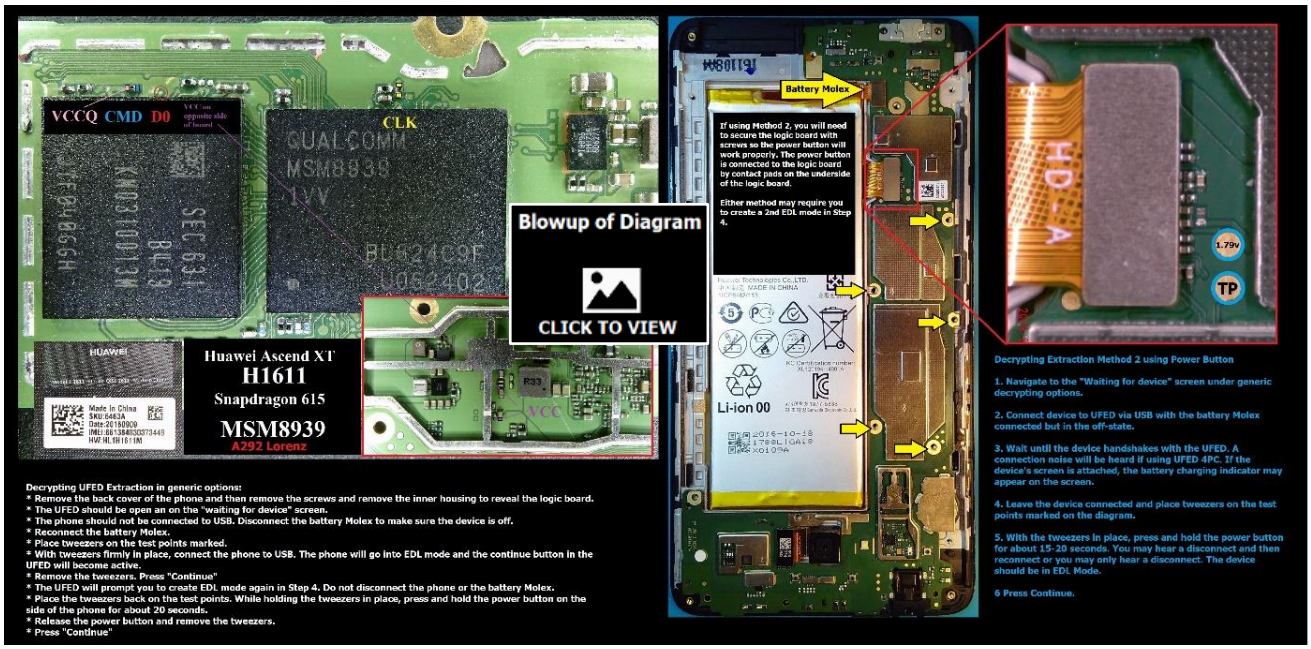
7.6 Huawei

Huawei phones have traditional round pads for test points connected to the same point on Qualcomm BGAs as the LG and other models. The test points I located were located in advantageous positions, improving extraction procedures, as was true for the H1611. Access to the logic board only required screw removal. On at least one device, I noticed the test point tested for 1.8 volts departed from the usually very low voltage of most phones. I don't think that characteristic is typical and the trigger was still voltage.

Test Point – low or moderate voltage

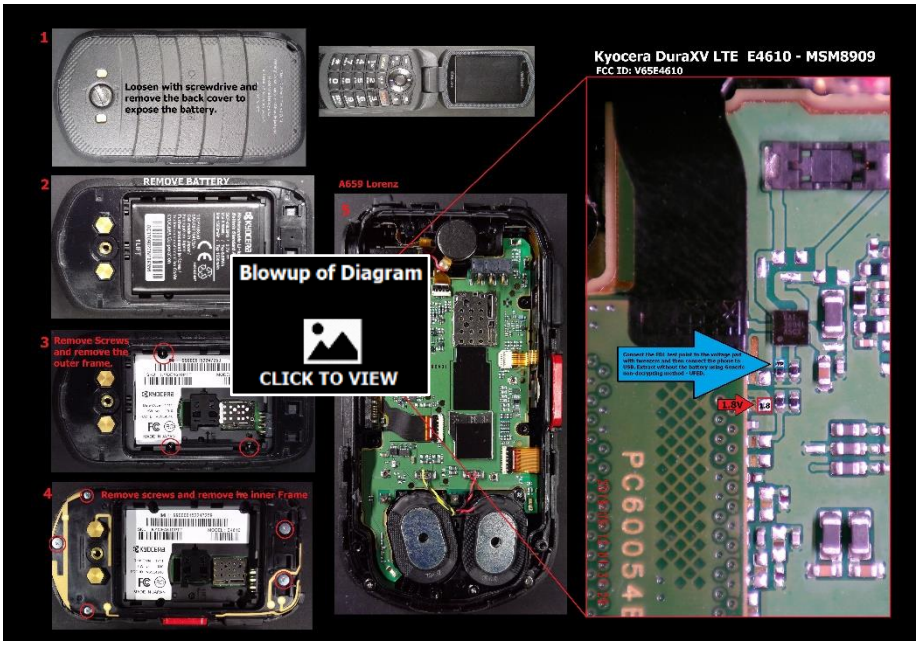
Trigger – 1.8 volts





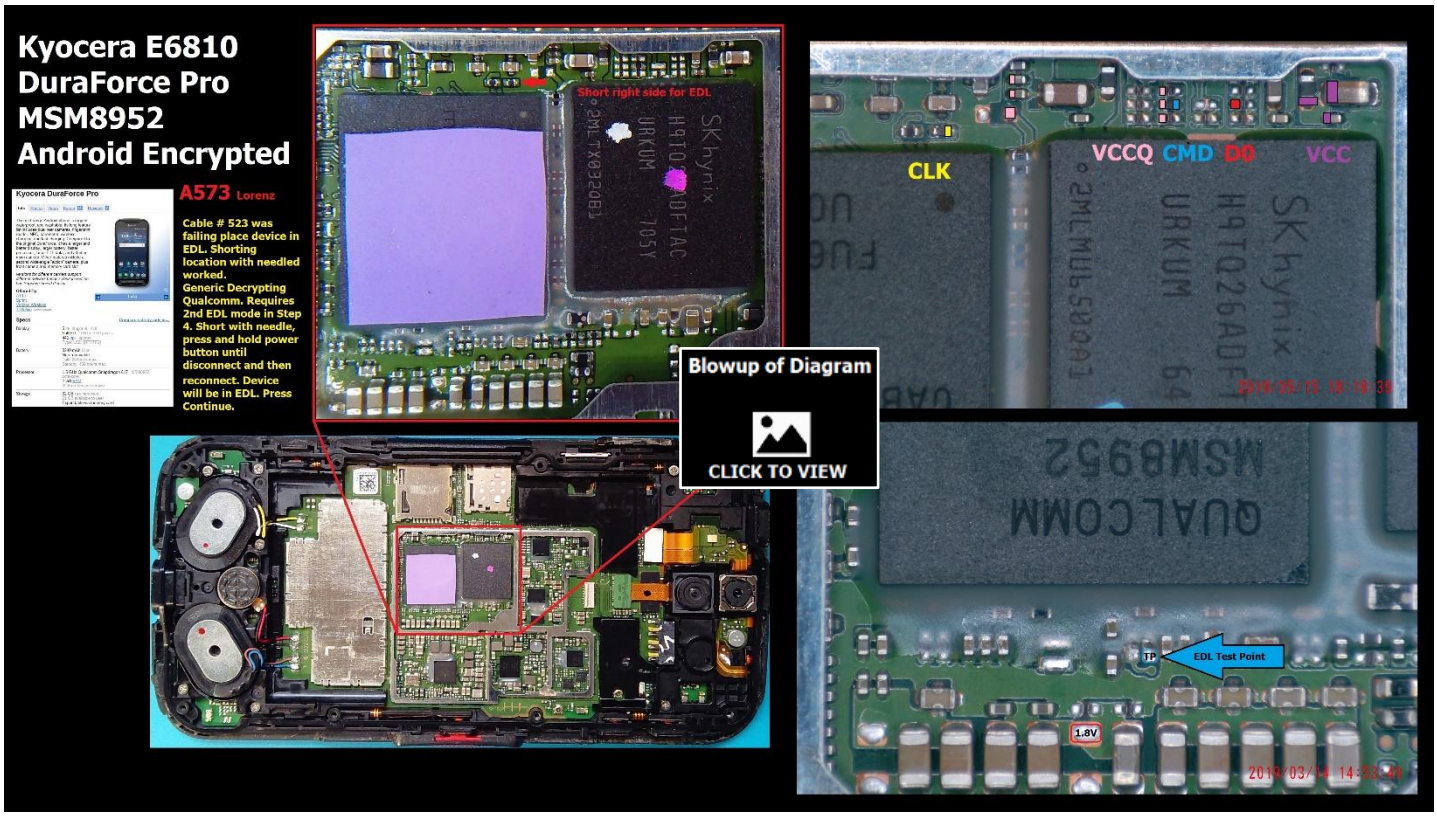
7.7 Kyocera

I have limited work with Kyocera devices but they are consistent with other devices in terms of the test point and the trigger. The design of some of the Dura models can make disassembly challenging but necessary as EDL cables and button combos did not work on these phones. The test points were not as easily identifiable as the location of the test point on the E6810 did not improve the logistics of extraction making it just as easy to create an eMMC fault. Locating the test point on the E4610 did significantly improve extraction for that device. The fault locations were located on the underside of the logic board and cover by heavy epoxy. I was able to trace the test point via continuity to an alternate location in which is appeared on the top side of the logic board, making a non-decrypting extraction fairly simple.



Test Point - low voltage

Trigger – 1.8 volts

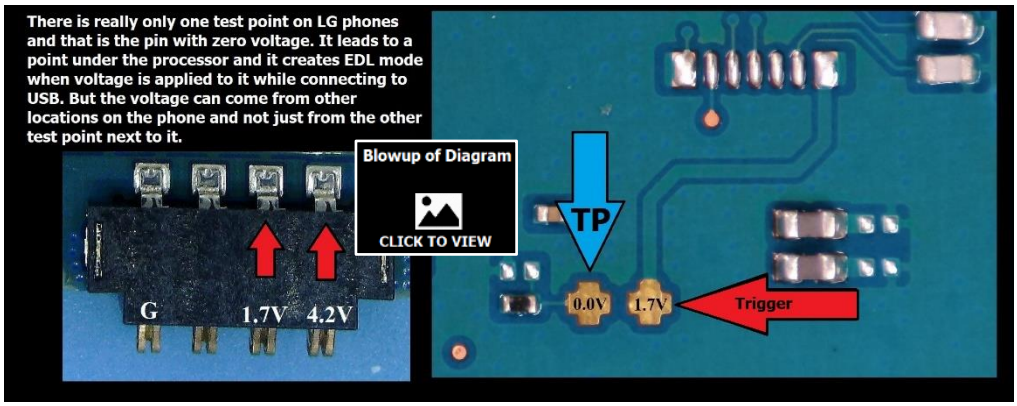


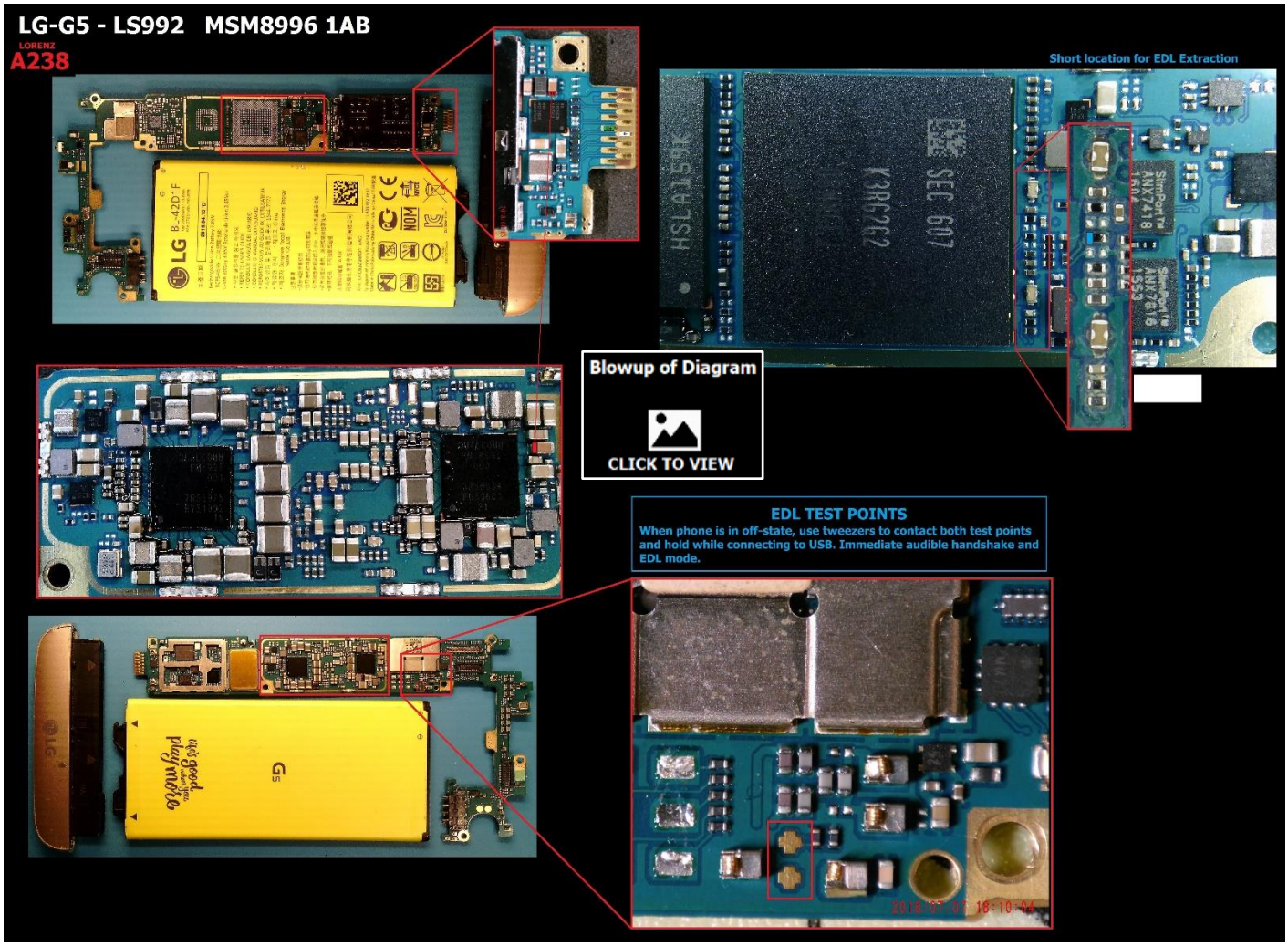
7.8 LG

Of course, LG has the most recognizable test point and trigger of all phones and served as a starting point for me to pin back to Qualcomm processors to locate the test point pin for other OEMs. The test point and trigger for most modern LG phones are in the shape of a cross and usually located in advantageous positions outside of heat shields and on the face-up side of the logic board when the back cover is removed. Disassembly of most LG devices is fairly simple with only screw removal required.

Test Point – no/low voltage

Trigger – 1.8 volts



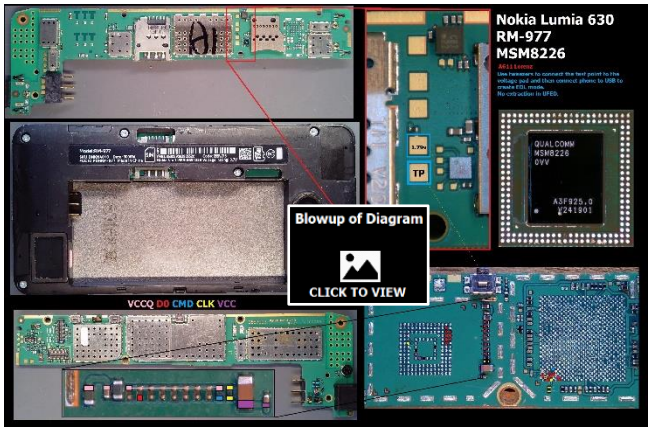


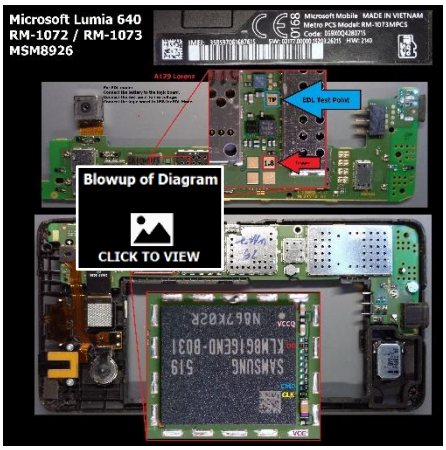
7.9 Microsoft / Nokia

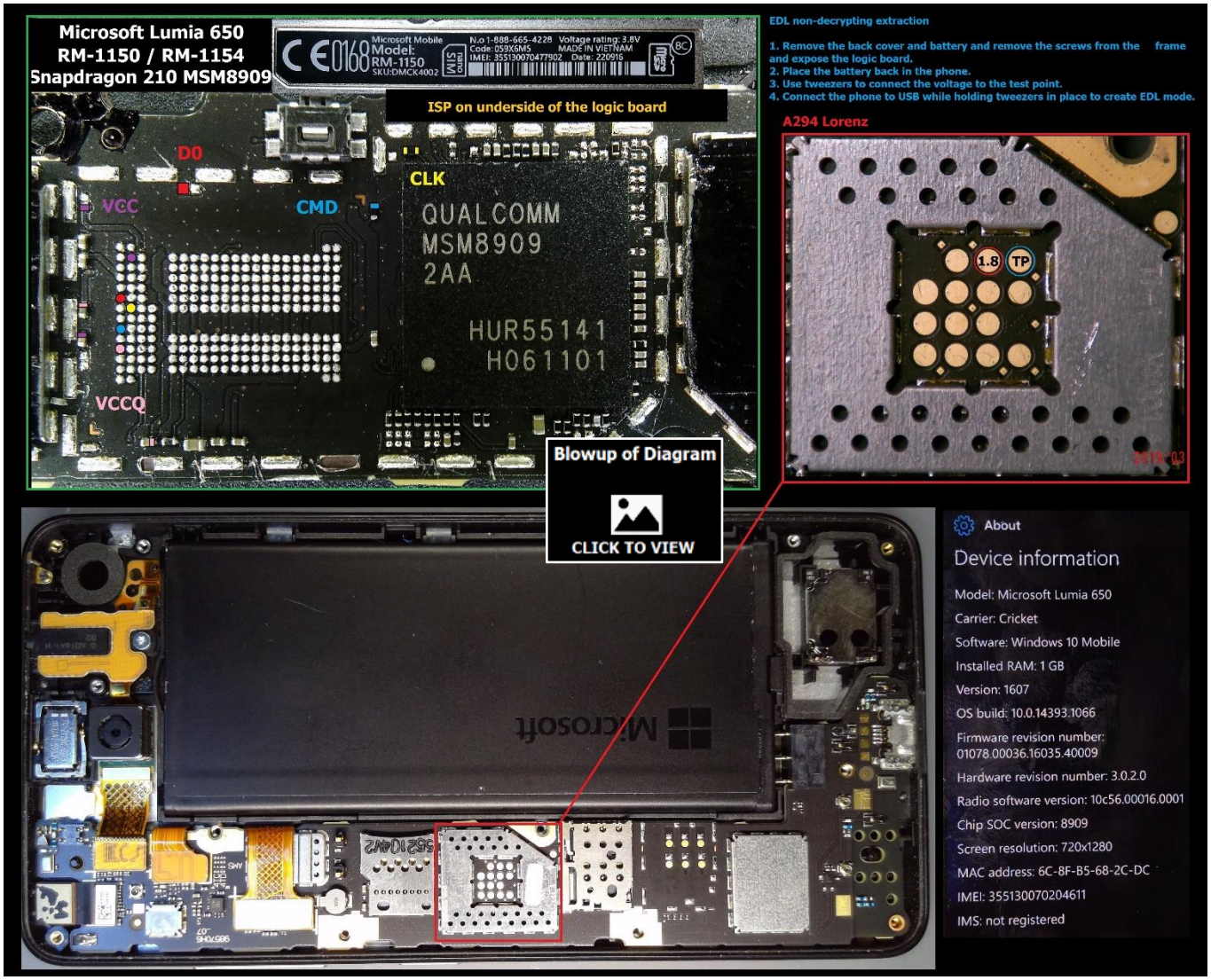
I only had a few of these phone to test and they follow the same patter as other OEMs with the test point being low or no voltage and the trigger usually 1.8 volts.

Test Point – no voltage

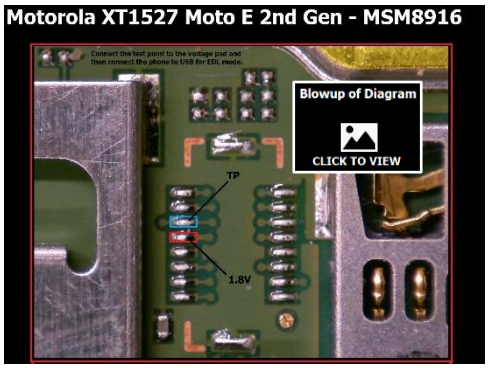
Trigger – 1.8 volts

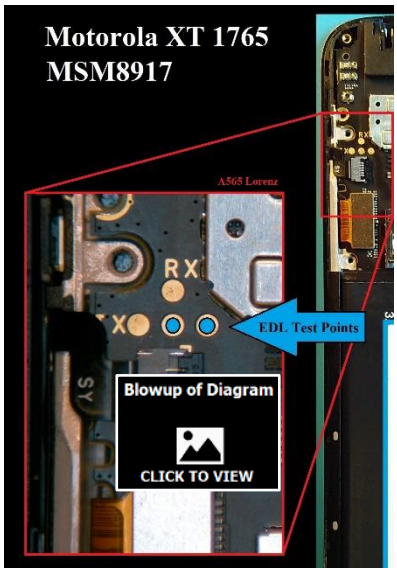






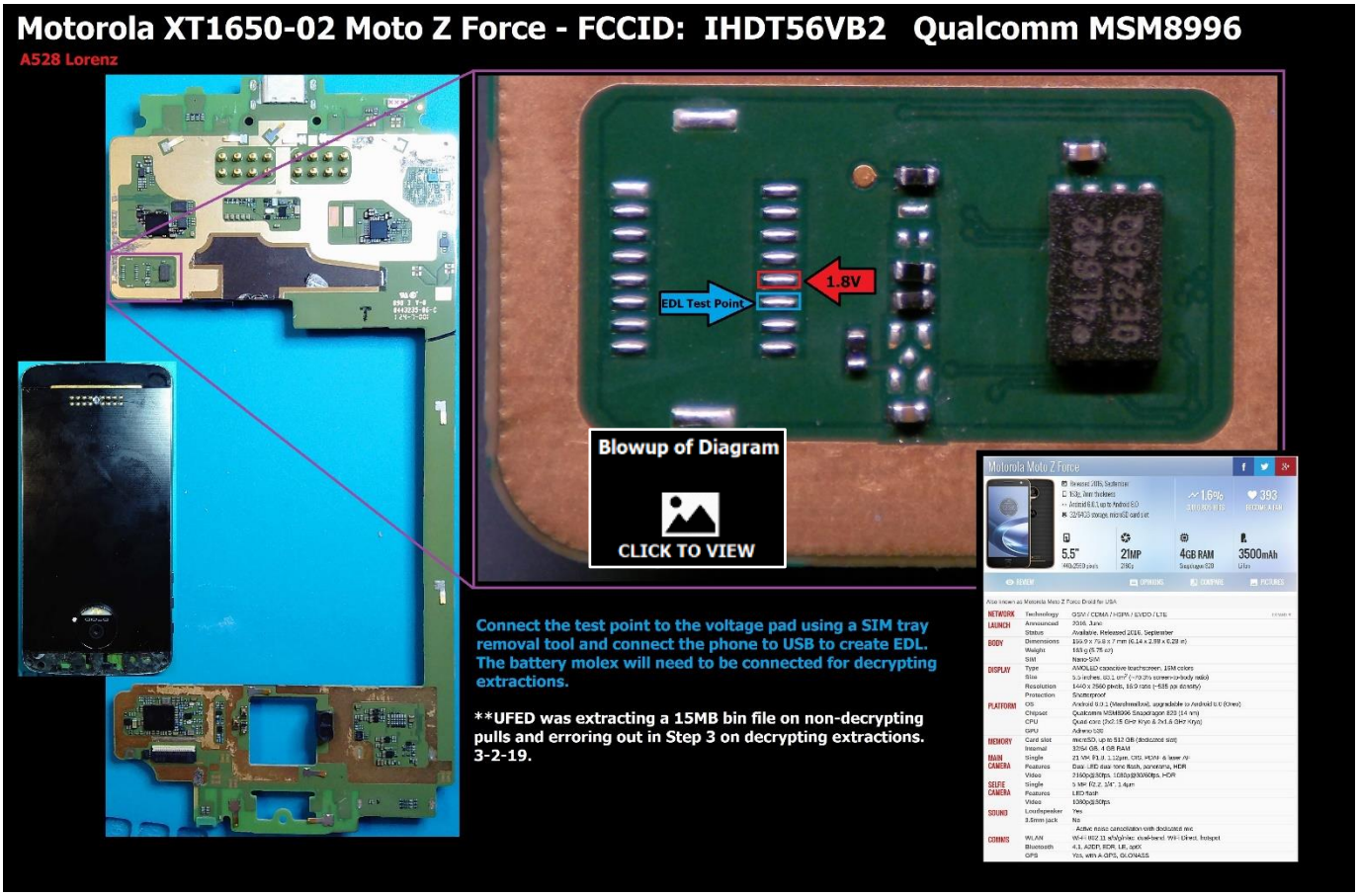
7.10 Motorola





Motorola follows the pattern of low voltage for the test point and 1.8 volts for the trigger. The known test points on Motorola phones are the traditional round pads. Those pads were usually placed in advantageous positions and, of course, the test point traces back to the same BGA pin under Qualcomm processors as LG and other models. After investigating Motorola models with no publicly known test points, I ended up reverse pinning from Qualcomm processors to discover the lesser-known EDL test point location on tiny un-mounted Molex pads. Once locating them on one device, others models were easy to locate and pinpoint based on testing for voltage of the particular location on the Molex pad that was the test point and the trigger. The test point is always located in the third position from the end with the trigger located in position 4. I noted that these locations are on newer Motorola phones including ones running the MSM8996 and on older models running back to the MSM8226.

Test Point – no voltage **Trigger** – 1.8 volts

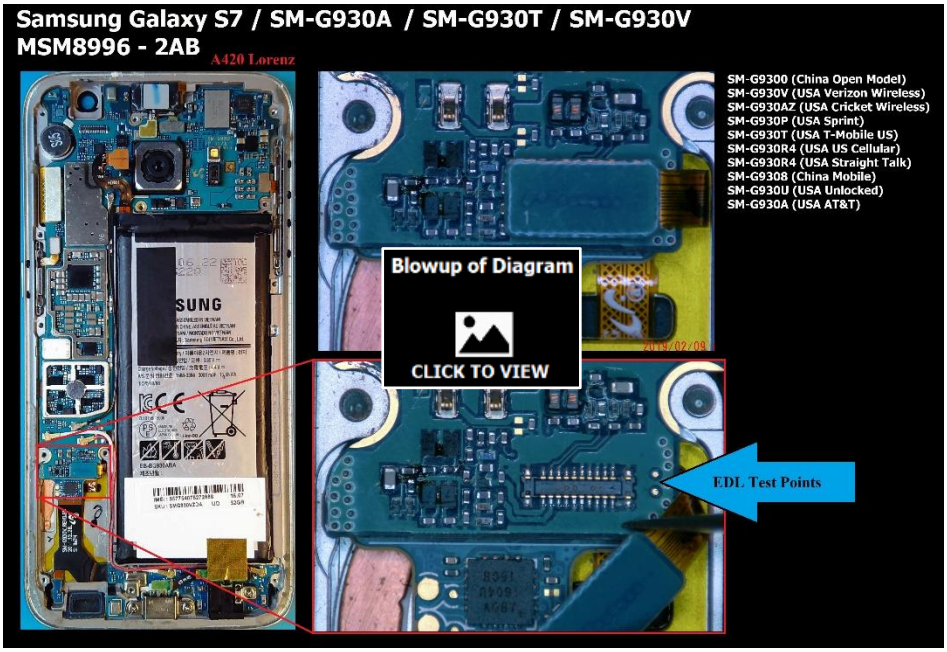


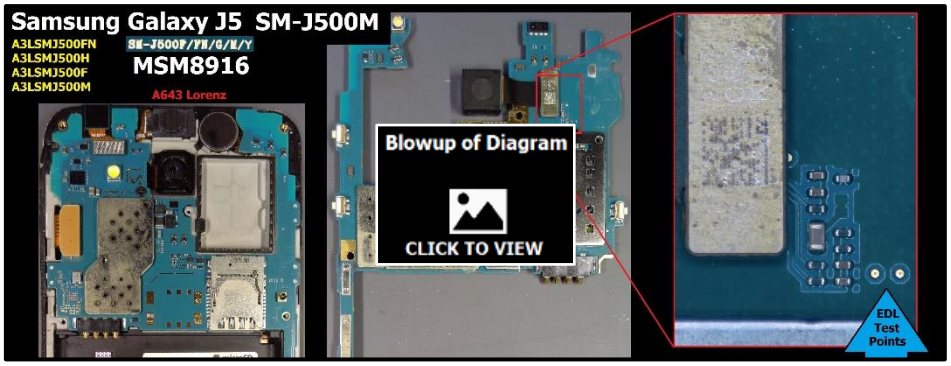
7.11 Samsung

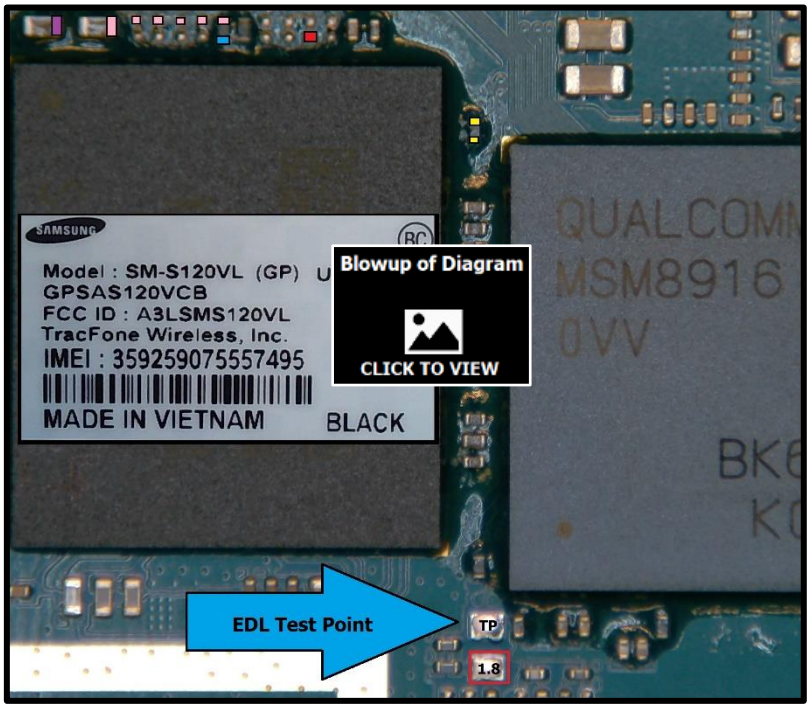
Test Point – no/low voltage

Trigger – 1.8 volts

Samsung devices do have EDL test points on older and newer models. Going back to devices running the MSM8916 processor, most of them have test points. However, these test points are located near the processor and eMMC, which usually requires flipping the logic board to access them. Models running the MSM8916 are never encrypted by default so the non-decrypting extraction works fine. The test point and the trigger on the older models are generally square pads following the same pattern of no voltage for the test point and 1.8 volts for the trigger. They trace back to the same location under the MSM8916 as other OEMs. There are few older Samsung devices in which the test point and the trigger are tiny round pads. These tiny pads test the same as the square pads and look like the tiny test point pads seen on the S7, S7 Edge, S7 Active, S8, and S9, all of which have a tiny round test point pad and trigger on the face up side of the logic board. Some Samsung models between the older ones running the 8916 and the S7 have test points that are not so easy to find and are less identifiable. Some S7 models are supported for EDL in the UFED but only via ADB. Non-decrypting extractions are possible for the S7 using test points but decrypting extractions have failed with my test. There is no current support for the MSM8998 for EDL so the S8 and S9 will not extract.

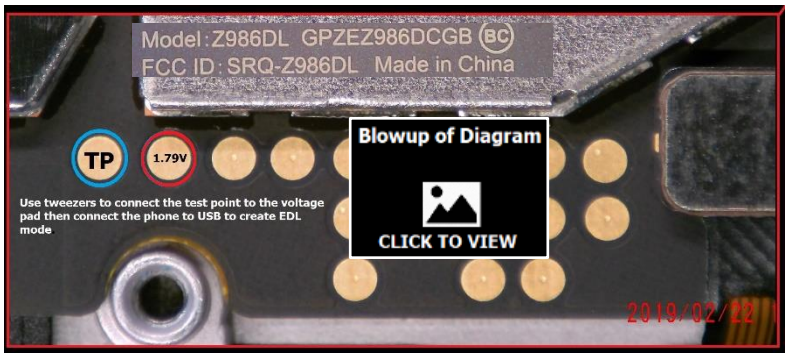


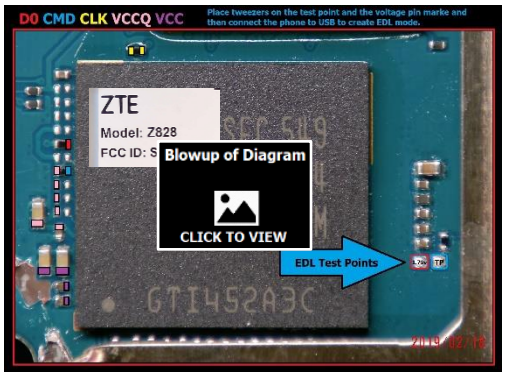


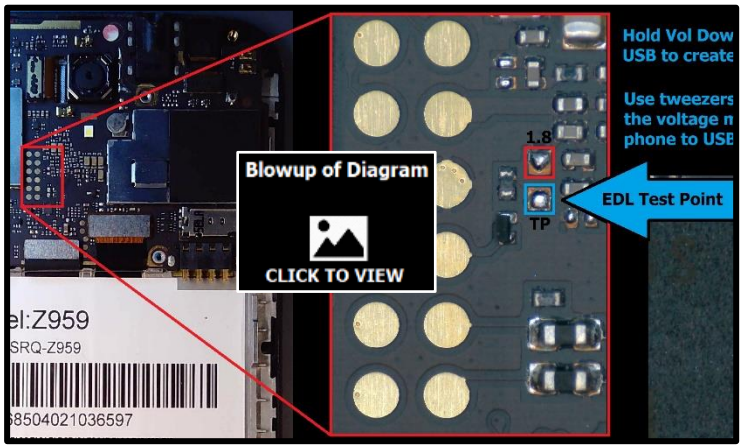


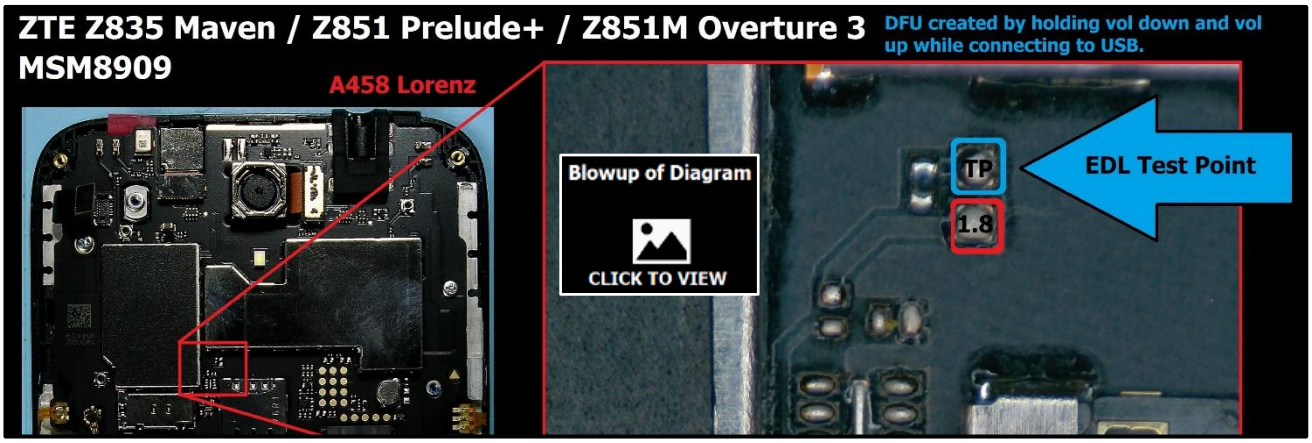
7.12 ZTE

Test Point – no/low voltage **Trigger** – 1.8 Volts. Many ZTE phones have test points. ZTE phones have a variety of different test point shapes and locations but all of the follow the rule of the test point having no voltage and the trigger usually being 1.8 volts. Some devices required higher voltage to trigger test point. Many of the ZTE devices have the traditional round pads located face up and the devices are generally easy to disassemble – only requiring a screwdriver. The test points do put examiners in a better place for extractions. Many ZTE phones can be placed in DFU or FTM mode, both of which can be exploited by Cellebrite without the need for disassembly. For EDL Mode to be created on some devices, test points are the best bet if an EDL cable is not supported.









8 EDL test point pinouts and videos

The following pinouts are for devices and variants which are both supported and not supported for extraction via EDL. Keep in mind that this list is based on phones and variants I could test. This list only scratches the surface of devices with Qualcomm processors with test points and which are exploitable via EDL Mode.

In addition to the links and diagrams I have provided in this paper, I have placed all of the diagrams, photos, and videos referenced in this paper inside the Mastering EDL Test Points folder on the Mobile Device Forensics and Analysis forum.

Happy Hunting…

Cellebrite Supports EDL extractions on any phone running these processors.

MSM8909 MSM8916 MSM8936 MSM8939 MSM8952

Cellebrite Supports EDL extractions select devices running these processors.

MSM8917 MSM8937 MSM8940 MSM8953 MSM8996

Vendor	Model	FCC	Processor	Inventory	Pinout	Video
Alcatel	4044C Cinglual Flip 2 / Quick Flip	2ACCJN012	MSM8909	**A590**	ISP-4044O	4044-TP
Alcatel	4044L Go Flip	2ACCJN012	MSM8909		ISP-4044O	4044-TP
Alcatel	4044M	2ACCJN014	MSM8909		ISP-4044O	4044-TP
Alcatel	4044O One Touch	2ACCJN012	MSM8909	**A652**	ISP-4044O	4044-TP
Alcatel	4044T Go Flip	2ACCJN010	MSM8909		ISP-4044O	4044-TP
Alcatel	4044V Go Flip	2ACCJN013	MSM8909		ISP-4044O	4044-TP
Alcatel	4044W	2ACCJN011	MSM8909		ISP-4044O	4044-TP
Alcatel	4060A	2ACCJB039	MSM8909	**A269**	ISP-4060A	02_5027B
Alcatel	4060A Ideal	2ACCJB039	MSM8909	**A270**	ISP-4060A	02_5027B
Alcatel	4060O Streak	2ACCJB039	MSM8909		ISP-4060A	02_5027B
Alcatel	4060S	2ACCJB039	MSM8909		ISP-4060A	02_5027B
Alcatel	4060W	2ACCJB039	MSM8909	**A297**	ISP-4060A	02_5027B
Alcatel	5017B One Touch Pixi 3 4.5	2ACCJB011	MSM8909	**A402**	ISP-5017B	
Alcatel	5027B Dawn	2ACCJB053	MSM8909	**A269**	ISP-4060A	02_5027B
Alcatel	5044C Verso	2ACCJB079	MSM8909	**A658**	ISP-5044R	29_5044R
Alcatel	5044R idealXCITE	2ACCJB079	MSM8909	**A490**	ISP-5044R	29_5044R
Alcatel	5044S	2ACCJB090	MSM8909		ISP-5044R	29_5044R
Alcatel	5046G	2ACCJBO75	MSM8909		EDL-A577VL	
Alcatel	5054N One Touch	2ACCJA008	MSM8909	**A494**	ISP-5054N	
Alcatel	5054S One Touch Pop 3 (5.5)	2ACCJA010	MSM8909		ISP-5054N	
Alcatel	5054W One Touch	2ACCJA010	MSM8909		ISP-5054N	
Alcatel	5056E	2ACCJB065	MSM8909		ISP-5056N	
Alcatel	5056N	2ACCJB062	MSM8909	**A427**	ISP-5056N	
Alcatel	5056O OneTouch Allura	2ACCJB043	MSM8909		ISP-5056N	
Alcatel	5056W	2ACCJB062	MSM8909		ISP-5056N	
Alcatel	5057M Pop 4 Plus CM	2ACCJB046	MSM8909		ISP-5056N	
Alcatel	6045B One Touch Idol 3	2ACCJN001	MSM8939		6045O-EDL	
Alcatel	6045I One Touch Idol 3	2ACCJN002	MSM8939		6045O-EDL	
Alcatel	6045K One Touch Idol 3		MSM8939		6045O-EDL	
Alcatel	6045O One Touch Idol 3	2ACCJN005	MSM8939	**A641**	6045O-EDL	
Alcatel	6045Y One Touch Idol 3		MSM8939		6045O-EDL	
Alcatel	6060S One Touch Idol 5S	2ACCJA024	MSM8953	**A625**	6060S-EDL	
Alcatel	6071W One Touch Idol 4S	2ACCJN009	MSM8996	**A505**	ISP-6071W	
Alcatel	7040N Fierce 2	RAD475	MSM8612	**A100**	ISP-7040N	
Alcatel	7040T One Touch Fierce 2	RAD475	MSM8612	**A100**	ISP-7040N	
Alcatel	7040T One Touch Pop Icon	RAD475	MSM8612	**A100**	ISP-7040N	
Alcatel	A405DL MyFlip	2ACCJN023	MSM8909	**A617**	ISP-4044O	405DL TP
Alcatel	A462C Pixi Eclipse	2ACCJB013	MSM8609	**A263**	ISP-462C	
Alcatel	A5044G	2ACCJB090	MSM8909		ISP-5044R	29_5044R
Alcatel	A520L Pop Nova	RAD535	MSM8916	**A303**	A521L-EDL-ISP	22_A521L
Alcatel	A521L One Touch Pop Star LTE 2	RAD534	MSM8916		A521L-EDL-ISP	22_A521L

Vendor	Model	FCC	Processor	Inventory	Pinout	Video
Alcatel	A564C One Touch Pop Icon	RAD476	MSM8612	**A307**	ISP-7040N	
Alcatel	A571VL Pixi Avion	2ACCJB027	MSM8909	**A269**	ISP-4060A	02_5027B
Alcatel	A574BL Raven	2ACCJB079	MSM8909	**A518**	ISP-5044R	29_5044R
Alcatel	A576BL	2ACCJBO75	MSM8909		EDL-A577VL	
Alcatel	A577VL	2ACCJB076	MSM8909	**A516**	EDL-A577VL	
Coolpad	3310A illumina	R38YL3310A	MSM8909-6	**A597**	EDL-3310A	
Coolpad	3622A Catalyst	R38YL3622A	MSM8909	**A430**	ISP-3622A	
Coolpad	3636A	R38YL3636A	MSM8917	**A567**	EDL-3636A	3636A
Coolpad	3701A REVVL Plus	R38YL3701A	MSM8953 3AB	**A522**	ISP-C3701A	
Google	G-2PW4100 Pixel	NM8G-2PW4100	MSM8996	**A585**	G2PW4100-EDL	Pixel-TP
HTC	0PCV100 Desire 510	NM80PCV100	MSM8926		ISP-0PCV200	
HTC	0PCV200 Desire 510	NM80PCV200	MSM8926	**1-23**	ISP-0PCV200	
HTC	0PCV220 Desire 510	NM80PCV220	MSM8926		ISP-0PCV200	
HTC	HTCD100LVW Desire 526	NM80PM3100	MSM8909	**1-12**	ISPD100L	
Huawei	H1611	QISH1611	MSM8939	**A578**	ISP-H1611	
Huawei	H1711 Ascend XT2	QISH1711	MSM8940	**A520**	EDL-H1711	
Huawei	H891L Pronto LTE		MSM8926	**A412**	ISP-891L	
Huawei	H892L Raven	QISH892L	MSM8926	**A301**	ISP-H892L	
Huawei	KII-L05 GR5	QISKII-L05	MSM8939	**A662**	EDL-KII-L05	
Huawei	RIO-AL00 G7 Plus		MSM8939	**A661**	RIO-L01	RIOL01-fault
Huawei	Y538 Union	QISY538	MSM8909	**1-16**	ISP-Y538	Y538-TP
Kyocera	E4610 DuraXV LTE	V65E4610	MSM8909	**A659**	EDL-E4610	31_E4610
Kyocera	E6810		MSM8952	**A573**	EDL-E6810	16_E6810
LG	H343 Risio	ZNFH345	MSM8916	**1-85**	ISP-LS665	
LG	H345	ZNFH345	MSM8916	**1-85**	ISP-LS665	
LG	H631		MSM8916	**A656**	H631	
LG	H634 G Stylo				H631	
LG	H811 G4	ZNFH811	MSM8992	**A429**	H811-EDL	
LG	H820 G5	ZNFH820	MSM8996		EDL-H860	
LG	H830 G5	ZNFH830	MSM8996		EDL-H860	
LG	H860 G5		MSM8996	**A507**	EDL-H860	
LG	H900 V10	ZNFH900	MSM8992		ISP-H901	
LG	H901 V10	ZNFH901	MSM8992	**A416**	ISP-H901	
LG	H910 V20	ZNFH910	MSM8996		EDL-H918	
LG	H918 V20	ZNFH918	MSM8996	**A527**	EDL-H918	
LG	H960 V10	ZNFH960	MSM8992		ISP-H901	
LG	H990 V20	ZNFH990	MSM8996		EDL-H918	
LG	H990DS V20	ZNFH990	MSM8996		EDL-H918	
LG	H990T	ZNFH990	MSM8996		EDL-H918	
LG	K120 K4		MSM8909	**A235**	ISP-L44VL	
LG	K120E K4	ZNFK120E	MSM8909	**A235**	ISP-L44VL	
LG	K371 Phoenix 2	ZNFK371	MSM8909	**A545**	ISP-EDL K371	
LG	K373 Escape 3	ZNFK373	MSM8909		ISP-EDL K371	
LG	K428	ZNFK428	MSM8909	**A228**	ISP-K428	
LG	L33L Sunset	ZNFH345	MSM8916	**1-85**	ISP-LS665	
LG	L44VL Rebel	ZNFL44VL	MSM8909	**A291**	ISP-L44VL	
LG	L52VL Treasure	ZNFL52VL	MSM8909		ISP-L51AL	
LG	L57BL	ZNFL57BL	MSM8909	**A413**	ISP-L57BL	
LG	L58VL Rebel 2 LTE	ZNFL58VL	MSM8909	**A512**	ISP-L57BL	
LG	L59BL Grace LTE	ZNFL59BL	MSM8917	**A513**	ISP-L59BL	
LG	L59VL	ZNFL59BL	MSM8917	**A513**	ISP-L59BL	
LG	LG52VL	ZNFL52VL	MSM8909	**A298**	ISP-L51AL	
LG	LGL51AL Treasure	ZNFK330	MSM8909	**A298**	ISP-L51AL	
LG	LGL61AL K10 TracFone	ZNFL61AL	MSM8916	**A228A**	ISP-L61AL	
LG	LGL62VL Premier	ZNFL62VL	MSM8916	**A464**	ISP-L61AL	
LG	LGL63BL Fiesta LTE	ZNFL63BL	MSM8917		ISP-L64VL	
LG	LGL64VL Fiesta LTE	ZNFL64VL	MSM8917	**A514**	ISP-L64VL	
LG	LGL81AL Stylo 2	ZNFL81AL	MSM8916	**A300**	ISP-L82VL	
LG	LGL82V Stylo 2	ZNFL82VL	MSM8916	**A300**	ISP-L82VL	
LG	LGL82VL Stylo 2	ZNFLGL82VL	MSM8916		ISP-L82VL	
LG	LGL84VL Stylo 3 LTE	ZNFL84VL	MSM8917		EDL L84VL	
LG	LGM320G (Canada)	ZNFM320G	MSM8917		ISP-L64VL	
LG	LGM322 X Charge Xfinity	ZNFM322	MSM8917		ISP-L64VL	
LG	LGM327 X charge	ZNFLGM327	MSM8917		ISP-L64VL	
LG	LGMP450 Stylo 3 Plus	ZNFTP450	MSM8940		EDL LGMP450	
LG	LGMP450 Stylo 3 Plus Titan	ZNFTP450	MSM8940		EDL LGMP450	
LG	LG-Q710CS Stylo 4	ZNFQ710CS	SDM450		EDL-Q710MS	

Vendor	Model	FCC	Processor	Inventory	Pinout	Video
LG	LG-Q710MS Stylo 4	LGFQ710MS	SDM450	**A644**	EDL-Q710MS	
LG	LG-Q710US Stylo 4	ZNFQ710US	SDM450		EDL-Q710MS	
LG	LML212VL	ZNFX210VPP	MSM8917		EDL-LM-X210	
LG	LM-X210APM	ZNFX210APM	MSM8917		EDL-LM-X210	
LG	LM-X210APM Phoenix 4	ZNFX210APM	MSM8917	**A593**	EDL-LM-X210	
LG	LM-X210CM		MSM8917		EDL-LM-X210	
LG	LM-X210CMR Risio 3		MSM8917		EDL-LM-X210	
LG	LM-X210MA	ZNFX210ULM	MSM8917		EDL-LM-X210	
LG	LM-X210ULM K8+	ZNFX210ULM	MSM8917		EDL-LM-X210	
LG	LM-X210VPP	ZNFX210VPP	MSM8917		EDL-LM-X210	
LG	LM-X210WM (K9)	ZNFX210APM	MSM8917		EDL-LM-X210	
LG	LM-X212TA Aristo 2 Plus	ZNFX212TA	MSM8917		EDL-LM-X210	
LG	LS-450 (K3)	ZNFLS450	MSM8909	**A399**	ISP-LS450	
LG	LS665	ZNFLS665	MSM8916	**A440**	ISP-LS665	
LG	LS675 Tribute 5	ZNFLS675	MSM8909	**1-52/1-53**	ISP-L51AL	
LG	LS676 Tribute HD	ZNFLS676	MSM8909	**A400**	ISP-LS676	LS676-LAF
LG	LS-696 Optimus Elite	ZNFLS696	MSM8909	**A122**	ISP-LS676	LS676-LAF
LG	LS770	ZNFLS770	MSM8916	**1-68**	LS770	
LG	LS777 Stylo 3	ZNFLS777	MSM8940	**A515**	ISP-LS777	
LG	LS992 G5	ZNFLS992	MSM8996	**A238**	EDL-LS992	
LG	LS997 V20 Titan	ZNFLS997	MSM8996		EDL-H918	
LG	M150	ZNFM150	MSM8909	**A419**	ISP-L57BL	
LG	M210 Aristo	ZNFM210	MSM8917		ISP-L57BL	
LG	M255 K20	ZNFM255	MSM8917	**A513**	ISP-L59BL	
LG	M257 Harmony	ZNFM255	MSM8917	**A513**	ISP-L59BL	
LG	M430 Stylo 3	ZNFM430	MSM8917 / MSM8940		ISP-LS777	
LG	MP260 K20 Plus	ZNFTP260	MSM8917	**A588**	ISP-L59BL	
LG	MS210 Aristo K8	ZNFM210	MSM8917	**A561**	TP-MS210	
LG	MS210UK Aristo	ZNFM210	MSM8917		TP-MS210	
LG	MS330 (K7)	ZNFK330	MSM8909	**A428**	ISP-L51AL	
LG	MS345 Leon LTE	ZNFH345	MSM8916	**A465**	ISP-LS665	
LG	MS428 K10	ZNFK428	MSM8909	**A228**	ISP-K428	
LG	MS550 Stylo 2 Plus	ZNFK550BN	MSM8937	**A568**	ISP-MS550	
LG	US992 G5	ZNFLS992	MSM8996	**A238**	EDL-LS992	
LG	US996 LRA V20	ZNFUS996	MSM8996	**A527**	EDL-H918	
LG	US996 Titan V20	ZNFUS996			EDL-H918	
LG	VS425 K4	ZNFVS425			ISP-L44VL	
LG	VS425PP Optimus Zone 3	ZNFVS425PP	MSM8909	**A279**	ISP-L44VL	
LG	VS501 K20	ZNFVS501	MSM8917		ISP-L59BL	
LG	VS987 G5	ZNFVS987	MSM8996		EDL-H860	
LG	VS995 V20	ZNFVS995	MSM8996		EDL-H918	
Microsoft	Lumia 640 (RM-1072)	PYATAA	MSM8926	**A129**	640-EDL-ISP	
Microsoft	Lumia 640 (RM-1073)	PYARM-1073	MSM8926		640-EDL-ISP	
Microsoft	Lumia 650 (RM-1150)	PYARM-1150	MSM8909		650-EDL-ISP	
Microsoft	Lumia 650 (RM-1154)	PYARM-1154	MSM8909	**A294**	650-EDL-ISP	
Motorola	XT1028 Moto G	IHDT56PF3	MSM8226		XT1045 EDL	
Motorola	XT1031 Moto G	IHDT56PF3	MSM8226	**A136**	XT1045 EDL	
Motorola	XT1032 Moto G	IHDT56PF3	MSM8226	**A631**	XT1032 - EDL	
Motorola	XT1045 Moto G	IHDT56PG1	MSM8926	**1-71**	XT1045 EDL	
Motorola	XT1095 Moto X 2nd Gen	IHDT56QA1	MSM8974-AC	**A162**	XT1095	
Motorola	XT1097 Moto X 2nd Gen	IHDT56QA1	MSM8974-AC	**A162**	XT1095	
Motorola	XT1524 Moto E 2nd Gen		MSM8916		XT1527-EDL	XT1527-TP
Motorola	XT1526 Moto E 2nd Gen	IHDT56QC1	MSM8916		XT1527-EDL	XT1527-TP
Motorola	XT1526 Moto E 2nd Gen LTE/CDMA	IHDT56QC8	MSM8916		XT1527-EDL	XT1527-TP
Motorola	XT1526 Moto E 2nd Gen LTE/CDMA	IHDT56QC7	MSM8916		XT1527-EDL	XT1527-TP
Motorola	XT1527 Moto E 2nd Gen	IHDT56QC1	MSM8916	**A606**	XT1527-EDL	XT1527-TP
Motorola	XT1528 Moto E	IHDT56QC8	MSM8916		XT1527-EDL	XT1527-TP
Motorola	XT1540 Moto G 3rd Gen	IHDT56QG1	MSM8916	**A608**	XT1540-EDL	
Motorola	XT1541 Moto G 3rd Gen		MSM8916		XT1540-EDL	
Motorola	XT1542 Moto G 3rd gen		MSM8916		XT1540-EDL	
Motorola	XT1543 Moto G 3rd Gen		MSM8916		XT1540-EDL	
Motorola	XT1548 Moto G 3rd Gen	IHDT56QG6	MSM8916		XT1540-EDL	
Motorola	XT1565 Droid Maxx 2	IHDT56UB1	MSM8939	**A272**	XT1565B EDL	
Motorola	XT1565B Droid Maxx 2	IHDT56UB1	MSM8939	**A607**	XT1565B EDL	
Motorola	XT1601 Moto G4 Play	IHDT56VD1	MSM8916		ISP-XT1609	

Vendor	Model	FCC	Processor	Inventory	Pinout	Video
Motorola	XT1602 Moto G4 Play	IHDT56VD3	MSM8916		ISP-XT1609	
Motorola	XT1604 Moto G4 Play	IHDT56VD5	MSM8916		ISP-XT1609	
Motorola	XT1607 Moto G4 Play	IHDT56VD6	MSM8916		ISP-XT1609	
Motorola	XT1609 Moto G4 Play	IHDT56VD4	MSM8916	**A499**	ISP-XT1609	
Motorola	XT1620 Moto G4	IHDT56VA3	MSM8952		XT1644-EDL	XT1644-TP
Motorola	XT1621 Moto G4	IHDT56VA4	MSM8952		XT1644-EDL	XT1644-TP
Motorola	XT1622 Moto G4	IHDT56VA2	MSM8952		XT1644-EDL	XT1644-TP
Motorola	XT1624 Moto G4	IHDT56VA1	MSM8952		XT1644-EDL	XT1644-TP
Motorola	XT1625 Moto G4		MSM8952		XT1644-EDL	XT1644-TP
Motorola	XT1626 Moto G4	IHDT56VA3	MSM8952		XT1644-EDL	XT1644-TP
Motorola	XT1635-01 Moto Z Play	IHDT56VC1	MSM8953 0AB	**A540**	EDL-XT1635-01	
Motorola	XT1640 Moto G4 Plus Dual	IHDT56VA3	MSM8952		XT1644-EDL	XT1644-TP
Motorola	XT1641 Moto G4 Plus	IHDT56VA4	MSM8952		XT1644-EDL	XT1644-TP
Motorola	XT1642 Moto G4 Plus	IHDT56VA2	MSM8952		XT1644-EDL	XT1644-TP
Motorola	XT1643 Moto G4 Pluss	IHDT56VA1	MSM8952		XT1644-EDL	XT1644-TP
Motorola	XT1644 Moto G4 Plus	IHDT56VA5	MSM8952	**A634**	XT1644-EDL	XT1644-TP
Motorola	XT1650-02 Moto Z Force	IHDT56VB2	MSM8996	**A528**	EDL-XT1650-02	
Motorola	XT1680 Moto G5 Plus		MSM8953		EDL-XT1685	
Motorola	XT1681 Moto G5 Plus		MSM8953		EDL-XT1685	
Motorola	XT1684 Moto G5 Plus		MSM8953		EDL-XT1685	
Motorola	XT1685 Moto G5 Plus		MSM8953		EDL-XT1685	
Motorola	XT1686 Moto G5 Plus		MSM8953		EDL-XT1685	
Motorola	XT1687 Moto G5 Plus		MSM8953 0AB	**A570**	EDL-XT1685	
Motorola	XT1710-01 Moto Z2 Play		MSM8953 Pro		EDL-XT1710-02	
Motorola	XT1710-02 Moto Z2 Play		MSM8953 Pro	**A648**	EDL-XT1710-02	
Motorola	XT1710-07 Moto Z2 Play Dual		MSM8953 Pro		EDL-XT1710-02	
Motorola	XT1710-08 Moto Z2 Play Dual		MSM8953 Pro		EDL-XT1710-02	
Motorola	XT1710-09 Moto Z2 Play Dual		MSM8953 Pro		EDL-XT1710-02	
Motorola	XT1710-10 Moto Z2 Play Dual		MSM8953 Pro		EDL-XT1710-02	
Motorola	XT1710-11 Moto Z2 Play Dual		MSM8953 Pro		EDL-XT1710-02	
Motorola	XT1765 Moto E4	IHDT56WC2	MSM8917	**A565**	ISP-XT176x	XT1765-SADB
motorola	XT1766 Moto E4 (USA)	IHDT56WC3	MSM8920	**A609**	ISP-XT176x	XT1765-SADB
Motorola	XT1767 Moto E4	IHDT56WC1	MSM8917	**A418**	ISP-XT176x	XT1765-SADB
Motorola	XT1768 Moto E4	IHDT56WC1	MSM8917		ISP-XT176x	XT1765-SADB
Motorola	XT1789 Moto Z2 Force	IHDT56WB2	MSM8998	**A649**	XT1789-EDL	
Motorola	XT1798-01 Moto Z2 Force		MSM8998		XT1789-EDL	
Motorola	XT1798-03 Moto Z2 Force		MSM8998		XT1789-EDL	
Motorola	XT1798-04 Moto Z2 Force		MSM8998		XT1789-EDL	
Motorola	XT1798-05 Moto Z2 Force		MSM8998		XT1789-EDL	
Motorola	XT1798-07 Moto Z2 Force		MSM8998		XT1789-EDL	
Motorola	XT1801 Moto G5S	IHDT56WH2	MSM8953		XT1806-EDL	
Motorola	XT1802 Moto G5S Plus	IHDT56WH3	MSM8953		XT1806-EDL	
Motorola	XT1803 Moto G5S Plus	IHDT56WH4	MSM8953		XT1806-EDL	
Motorola	XT1804 Moto G5 Plus	IDHT56WH5	MSM8953		XT1806-EDL	
Motorola	XT1806 Moto G5S Plus	IHDT56WH1	MSM8953	**A633**	XT1806-EDL	
Motorola	XT1921-2 Moto E5 Cruise	IHDT56XC4	MSM8920		EDL-XT1921-5	
Motorola	XT1921-3 Moto E5 Play	IHDT56XC2	MSM8920	**A596**	EDL-XT1921-5	
Motorola	XT1921-5 Moto E5 Play	IHDT56XC2	MSM8920	**A554**	EDL-XT1921-5	
Motorola	XT1922-7 Moto G6 Play	IHDT56XB1	MSM8920	**A557**	XT1922-7	
Motorola	XT1924-1 Moto E5 Plus	IHDT56XA4	MSM8917		XT1924-7	
Motorola	XT1924-2 Moto E5 Plus	IHDT56XA4	MSM8917		XT1924-7	
Motorola	XT1924-3 Moto E5 Plus	IHDT56XA6	MSM8937		XT1924-7	
Motorola	XT1924-4 Moto E5 Plus	IHDT56XA5	MSM8917		XT1924-7	
Motorola	XT1924-5 Moto E5 Plus	IHDT56XA5	MSM8917		XT1924-7	
Motorola	XT1924-6 Moto E5 Plus	IHDT56XA1	MSM8940		XT1924-7	
Motorola	XT1924-7 Moto E5 Plus	IHDT56XA2	MSM8940	**A610**	XT1924-7	
Motorola	XT1924-8 Moto E5 Plus	IHDT56XA1	MSM8940		XT1924-7	
Motorola	XT1925-1 Moto G6	IHDT56XD5	SDM450		XT1925DL	
Motorola	XT1925-12 Moto G6	IHDT56XD1	SDM450		XT1925DL	
Motorola	XT1925-2 Moto G6	IHDT56XD5	SDM450		XT1925DL	
Motorola	XT1925-3 Moto G6	IHDT56XD6	SDM450		XT1925DL	
Motorola	XT1925-4 Moto G6	IHDT56XD4	SDM450		XT1925DL	
Motorola	XT1925-5 Moto G6	IHDT56XD4	SDM450		XT1925DL	
Motorola	XT1925-6 Moto G6	IHDT56XD1	SDM450		XT1925DL	
Motorola	XT1925DL Moto G6	IHDT56XD1	SDM450	**A595**	XT1925DL	
Nokia	Lumia 630 (RM-977)	PDNRM-977	MSM8926	**A611**	Lumia 630	
Samsung	SGH-i537 Galaxy S4 Active	A3LSGHI537	APQ8064T	**1-86**	SGH-i537	

Vendor	Model	FCC	Processor	Inventory	Pinout	Video
Samsung	SM-A300F Galaxy A3	A3LSMA300F	MSM8916	**A666**	A300FU	A300FU-fault
Samsung	SM-A300FU Galaxy A3 LTE	A3LSMA300FU	MSM8916		A300FU	A300FU-fault
Samsung	SM-G360AZ Galaxy Core Prime	A3LSMG360AZ	MSM8916		ISP-G360T1	G360T1-TP
Samsung	SM-G360F Galaxy Core Prime	A3LSMG360FY	MSM8916		ISP-G360T1	G360T1-TP
Samsung	SM-G360G Galaxy Core Prime LTE	A3LSMG360G	MSM8916		ISP-G360T1	G360T1-TP
Samsung	SM-G360M Galaxy Core Prime	A3LSMG360M	MSM8916		ISP-G360T1	G360T1-TP
Samsung	SM-G360P Galaxy Core Prime	A3LSMG360P	MSM8916	**1-88**	SM-S820L	
Samsung	SM-G360T Galaxy Core Prime	A3LSMG360T	MSM8916		ISP-G360T1	G360T1-TP
Samsung	SM-G360T1 Galaxy Core Prime	A3LSMG360T	MSM8916	**A446**	ISP-G360T1	G360T1-TP
Samsung	SM-G360V Galaxy Core Prime	A3LSMG360V	MSM8916	**1-88**	SM-S820L	S820L-TP
Samsung	SM-G530A Galaxy Grand Prime	A3LSMG530AZ	MSM8916		ISP-530P	530P-TP
Samsung	SM-G530AZ Galaxy Grand Prime	A3LSMG530AZ	MSM8916		ISP-530P	530P-TP
Samsung	SM-G530P Galaxy Grand Prime	A3LSMG530P	MSM8916	**1-41**	ISP-530P	530P-TP
Samsung	SM-G530R4 Galaxy Grand Prime	A3LSMG530R4	MSM8916		SM-S920L	S920L-TP
Samsung	SM-G530T Galaxy Grand Prime	A3LSMG530T	MSM8916		ISP-530P	530P-TP
Samsung	SM-G891A Galaxy S7 Active	A3LSMG891A	MSM8996	**A548**	EDL-G891A	
Samsung	SM-G9300 Galaxy S7	A3LSMG9300	MSM8996		EDL-G930A	
Samsung	SM-G9308 Galaxy S7 Duos		MSM8996		EDL-G930A	
Samsung	SM-G930A Galaxy S7	A3LSMG930US	MSM8996	**A420**	EDL-G930A	
Samsung	SM-G930P Galaxy S7	A3LSMG930US	MSM8996		EDL-G930A	
Samsung	SM-G930R4 Galaxy S7	A3LSMG930US	MSM8996		EDL-G930A	
Samsung	SM-G930T Galaxy S7	A3LSMG930US	MSM8996	**A551**	EDL-G930A	
Samsung	SM-G930U Galaxy S7		MSM8996	**A576**	EDL-G930A	
Samsung	SM-G930V Galaxy S7	A3LSMG930US	MSM8996	**A549**	EDL-G930A	
Samsung	SM-G935A Galaxy S7 Edge	A3LSMG935US	MSM8996	**A525**	EDL-G935A	
Samsung	SM-G935P Galaxy S7 Edge	A3LSMG935US	MSM8996		EDL-G935A	
Samsung	SM-G935R4 Galaxy S7 edge	A3LSMG935US	MSM8996		EDL-G935A	
Samsung	SM-G935T Galaxy S7 Edge	A3LSMG935US	MSM8996		EDL-G935A	
Samsung	SM-G935V Galaxy S7 Edge	A3LSMG935US	MSM8996	**A587**	EDL-G935A	
Samsung	SM-G950U Galaxy S8	A3LSMG950U	MSM8998	**A575**	EDL-G950U	
Samsung	SM-G960k Galaxy S9	A3LSMG960KOR	MSM8998		EDL-G960U	
Samsung	SM-G960U Galaxy S9	A3LSMG960U	MSM8998	**A583**	EDL-G960U	
Samsung	SM-J100VPP Galaxy J1	A3LSMJ100VPP	MSM8916	**A637**	SM-J100VPP	
Samsung	SM-J320FN Galaxy J3 2016	A3LSMJ320FN	MSM8916		J320P-ISP-EDL	
Samsung	SM-J320P Galaxy J3	A3LSMJ320P	MSM8916	**A624**	J320P-ISP-EDL	
Samsung	SM-J320R4 Galaxy J3 2016	A3LSMJ320R4	MSM8916 / SC9830		J320P-ISP-EDL	
Samsung	SM-J320VL Galaxy J3	A3LSMJ320VPP	MSM8916		J320P-ISP-EDL	
Samsung	SM-J320VPP Galaxy J3 2016	A3LSMJ320VPP	MSM8916		J320P-ISP-EDL	
Samsung	SM-J327P (2016.10.21)	A3LSMJ327P	MSM8917 / MSM8937	**A417**	ISP-J327P	
Samsung	SM-J327R4 Galaxy J3 2017	A3LSMJ327R4	MSM8917		ISP-J327P	
Samsung	SM-J327VPP Galaxy J3 2017	A3LSMJ327P	MSM8917	**A589**	ISP-J327P	
Samsung	SM-J500F Galaxy J5	A3LSMJ500F	MSM8916		EDL-J500M	J500M-TP
Samsung	SM-J500FN Galaxy J5	A3LSMJ500FN	MSM8916		EDL-J500M	J500M-TP
Samsung	SM-J500H Galaxy J5	A3LSMJ500H	MSM8916		EDL-J500M	J500M-TP
Samsung	SM-J500M Galaxy J5	A3LSMJ500M	MSM8916	**A643**	EDL-J500M	J500M-TP
Samsung	SM-J500Y Galaxy J5	A3LSMJ500Y	MSM8916		EDL-J500M	J500M-TP
Samsung	SM-J700P Galaxy J7	A3LSMJ700P	MSM8929	**A289**	SM-J700P	
Samsung	SM-J727P Galaxy J7 (2017)	A3LSMJ727P	MSM8953 1AB	**A530**	EDL-J727P	
Samsung	SM-J727R4 Galaxy J7 2017	A3LSMJ727R4	MSM8953		EDL-J727P	
Samsung	SM-J727V Galaxy J7 (2017)	A3LSMJ727V	MSM8953		EDL-J727P	
Samsung	SM-J727VL Galaxy J7 (2017)	A3LSMJ727VL	MSM8953		EDL-J727P	
Samsung	SM-S120VL Galaxy Luna	A3LSMS120VL	MSM8916	**A529**	EDL-S120VL	
Samsung	SM-S327VL Galaxy J3 (2016.12.08)	A3LSMS327VL	MSM8917	**A536**	ISP-J327P	
Samsung	SM-S327VL Galaxy J3 (2016.12.08)	A3LSMS327VL	MSM8917	**A581**	ISP-J327P	
Samsung	SM-S820L Galaxy Core Prime	A3LSMG360V	MSM8916	**A497**	SM-S820L	S820L-TP
Samsung	SM-S920L Galaxy Grand Prime	A3LSMS920L	MSM8916	**A571**	SM-S920L	S920L-TP
Sharp	306SH Aquos Crystal	APYHRO00204	MSM8926	**A613**	306SH-ISP	
Xiaomi	Mi 5X		MSM8953	**A555**	TP-Xiaomi Mi5x	
ZTE	N818S Qlink		MSM8909	**A667**	N818S-EDL	N818S-DFU
ZTE	N9130 Speed	SRQ-N9130	MSM8916	**1-08**	N9130	
ZTE	N9131 Tempo	SRQ-N9131	MSM8909	**A677**	N9131	
ZTE	N9136 Prestige 2	SRQ-N9136	MSM8909	**A553**	EDL-N9136	N9136-FTM
ZTE	N9137 (Tempo X)	SRQ-N9137	MSM8909	**A492**	EDL-N9137	
ZTE	N9518 Warp	SRQ-ZTEN9518	MSM8916	**1-50**	N9518-TP	
ZTE	N9519 Warp 7	SRQ-ZTEN9519	MSM8916		N9519 - TP	
ZTE	N9521	SRQ-ZTE9521	MSM8916		N9521	

Vendor	Model	FCC	Processor	Inventory	Pinout	Video
ZTE	N9560 Max XL	SRQ-ZTEN9560	MSM8940	**A559**	EDL-N9560	
ZTE	Z233V	SRQ-Z233VL	MSM8909		ISP-Z320	
ZTE	Z233VL Cymbal-C LTE	SRQ-Z233VL	MSM8909		ISP-Z320	
ZTE	Z320 Cymbal	SRQ-Z320	MSM8909	**A282**	ISP-Z320	
ZTE	Z557BL ZFive G	SRQ-Z557BL	MSM8909	**A532**	ISP-Z852	
ZTE	Z558VL Zfive C LTE	SRQ-Z558VL	MSM8909		ISP-Z852	
ZTE	Z716BL Citrine	SRQ-Z716BL	MSM8909	**A260**	ISP-TP-Z716BL	
ZTE	Z717BL Citrine LTE	SRQ-Z716BL	MSM8909		ISP-TP-Z716BL	
ZTE	Z717VL Citrine	SRQ-Z717VL	MSM8909	**A461**	ISP-TP-Z716BL	
ZTE	Z718TL Jasper	SRQ-Z718TL	MSM8909		ISP-TP-Z716BL	
ZTE	Z718TL Jasper LTE	SRQ-Z718TL	MSM8909	**A436**	ISP-TP-Z716BL	
ZTE	Z719DL Zmax One	SRQ-Z719DL	MSM8909	**A526**	EDL-Z719	
ZTE	Z753G Paragon	SRQ-Z752C	MSM8210	**A306**	Z755	
ZTE	Z755 Sonata 2	SRQ-Z755	MSM8210	**1-94**	Z755	
ZTE	Z798BL Majesty PRO LTE	SRQ-Z798BL	MSM8909		ISP-Z799VL	Z799VL-TP
ZTE	Z799VL Majesty PRO LTE	SRQ-Z799VL	MSM8909	**A577**	ISP-Z799VL	Z799VL-TP
ZTE	Z812	SRQ-Z812	MSM8916	**1-95**	ISP-Z812	
ZTE	Z813	SRQ-Z812	MSM8916		ISP-Z812	
ZTE	Z815 Fanfare 2	SRQ-Z815	MSM8909	**A398**	Z815-ISP-EDL	
ZTE	Z828 Avid Plus	SRQ-Z828	MSM8909	**A431**	Z828-ISP-TP	
ZTE	Z831 Maven 2	SRQ-Z831	MSM8909	**A234**	Z832-ISP-EDL	
ZTE	Z832 Sonata 3	SRQ-Z831	MSM8909	**A234**	Z832-ISP-EDL	
ZTE	Z833 Avid Trio	SRQ-Z833	MSM8917		ISP-Z836BL	
ZTE	Z835 Maven 3	SRQ-Z835	MSM8909	**A458**	ISP-Z835	
ZTE	Z836BL Zfive 2 LTE	SRQ-Z836BL	MSM8917		ISP-Z836BL	
ZTE	Z837VL Zfive 2 LTE	SRQ-Z837VL	MSM8917	**A560**	ISP-Z836BL	
ZTE	Z839 Blade Vantage		MSM8909	**A558**	EDL-Z839	Z839-DFU
ZTE	Z851 Prelude+	SRQ-Z835	MSM8909		ISP-Z835	
ZTE	Z851M Overture 3	SRQ-Z835	MSM8909		ISP-Z835	
ZTE	Z852 Fanfare 3	SRQ-Z852	MSM8909	**A533**	ISP-Z852	
ZTE	Z861BL ZFive L LTE	SRQ-Z861BL	MSM8909	**A623**	Z861BL-EDL	
ZTE	Z862VL ZFive L LTE	SRQ-Z862VL	MSM8909		Z861BL-EDL	
ZTE	Z899VL Majesty Pro Plus LTE	SRQ-Z799VL	MSM8909	**A519**	ISP-Z799VL	Z899VL-TP
ZTE	Z916BL ZMAX GRAND LTE	SRQ-Z916BL	MSM8909		Z959 ISP EDL	
ZTE	Z917VL ZMAX Champ LTE	SRQ-Z917VL	MSM8909	**A628**	Z959 ISP EDL	
ZTE	Z955L Zmax 2	SRQ-955L	MSM8916	**A627**	Z958-EDL-ISP	
ZTE	Z958 Zmax 2	SRQ-Z958	MSM8916	**A632**	Z958-EDL-ISP	
ZTE	Z959 Grand X3	SRQ-Z959	MSM8909	**1-90**	Z959 ISP EDL	
ZTE	Z965 Blade X	SRQ-Z965	MSM8917	**A621**	EDL-Z965	
ZTE	Z971 Blade Spark	SRQ-Z971	MSM8917	**A489**	ISP-Z971	Z971-523
ZTE	Z981 ZMAX Pro	SRQ-Z981	MSM8952-2AB	**A236**	Z981-ISP-EDL	
ZTE	Z982 Blade Z Max	SRQ-Z982	MSM8940	**A642**	Z982-EDL	
ZTE	Z986DL Max Blue LTE	SRQ-Z986DL	MSM8953 0AB	**A622**	Z986DL-IPS-TP	Z986DL-TP
ZTE	Z987 Grand X Max+	SRQ-Z87	MSM8926	**1-21**	Z987-ISP-EDL	

9 Videos of all extractions and procedures

The following is a list and description of all videos related to this document. Some of the videos are raw and have not been edited. A brief description of each video is provided as well as the length of each video.

Video	OEM	Description	Length
01_XT1644 – Final Edit	Motorola	Motorola XT1644 Moto G4 Plus Decrypting extraction using test point	2:38
01L_791 Cable 523 Extraction	ZTE	UFED 7.8 using cable # 523 for decrypting extraction	1:53
02_Alcatel 5027B Test Point Extraction	Alcatel	Alcatel One Touch phones. How to use extraction flow PDF and generic options with variants and devices not directly supported under their device profile	3:46
02L_Z839 DFU Mode Extraction	ZTE	Z839 Blade DFU non-decrypting extraction with UFED 7.8	2:08
03_Samsung SM-J500M	Samsung	Samsung J500M using test points on phone. Phone has legacy bootloader support in UFED and suggested profile is for EDL. Video shows non-decrypting extraction in UFED using test points	2:30
03L_N9136 FTM Extraction	ZTE	N9136 Decrypting extraction in Generic options using FTM – UFED 7.8	3:51
04_XT1527	Motorola	Motorola XT157 – Has lock bypass support for bootloader with security patch up to May 2017 in UFED. Suggested profile is for Decrypting EDL. Video shows non-decrypting extraction under generic options in UFED using test points.	0:40
04L_MS330 Test Point Extraction	LG	LG LGL51AL non-decrypting Generic extraction using test point and tweezers – UFED 7.8	2:35
05_Z852 – Non-decrypting	ZTE	ZTE Z852 / N9137 / Z755BL non-decrypting extraction using test point. Z852 and N9137 show decrypting extraction under their profile in the UFED using button combination. Z557BL show physical (if rooted) under its profile and suggested profile of EDL Decrypting. These 3 phones are not encrypted by default. Generic non-decrypting extraction video using a test point.	0:57
06_Z852 Decrypting	ZTE	ZTE Z852 / N9137 / Z755BL non-decrypting extraction using test point. Z852 and N9137 show decrypting extraction under their profile in the UFED using button combination. Z557BL show physical (if rooted) under its profile and suggested profile of EDL Decrypting. These 3 phones are not encrypted by default. Generic decrypting extraction in the UFED using test point	1:53
07_405DL – Cable X Tweezers	Alcatel	Alcatel 405DL EDL mode using Cable X and accessing test point without disassembly.	0.17
07L_Coolpad 3636A Decrypting EDL - boot to charge-only	Coolpad	Coolpad 3636A Decrypting EDL extraction under generic options. Using button combos to create EDL after charging indicator appears on screen. Corrects issue with applying decrypting bootloader on phones that boot to charge-only mode.	3:27

Video	OEM	Description	Length
08_523 Alcatel No Batt Boot	Alcatel	Alcatel 405DL EDL mode using Cable # 523 with no battery between Alcatel bootloop. Some Alcatel phones will continually try to boot if connected to USB with no battery. Video demos easy way to use Cable # 523 for EDL mode between boots.	0.30
08L_Oppo R9S - Cable 523 Decrypting Extraction	Oppo	Oppo R9S decrypting extraction under Generic options. Using cable # 523 to create EDL mode after charging indicator appears on screen. Corrects issue with applying decrypting bootloader on phones that boot to charge-only mode.	3:32
09_4044O Cable X Test Point	Alcatel	Alcatel 4044O showing Cable X to create EDL mode using test point	0.15
10_Alcatel Cable X - Voltage Test - EDL	Alcatel	Alcatel phones have a test point that test for voltage and are triggered by ground. Video shows how to test for voltage and create EDL using Cable X	4:48
11_ZTE Z986DL	ZTE	Using Cable X to demonstrate EDL test point on ZTE Z986DL phone. Using Cable X to apply voltage to the test point to create EDL	2:40
12_Kyocera E6810 eMMC Faults Cable X	Kyocera	Using Cable X to demonstrate eMMC Fault creation and extraction on Kyocera E6810.	4:55
13_Alcatel A405DL 523 Extraction	Alcatel	Using Cable # 523 for Alcatel 405DL extraction under generic non-decrypting options	1:26
14_14_SM-G530P EDL Extraction	Samsung	SM-G530P / SM-G530T / SM-G530A / SM-G530AZ	1:26
15_Cable X Demo Prototype 2	ZTE	Demo of Cable X for FTM / DFU / Test Points / Cable 523 / eMMC Faults	22:13
16_Kyocera E6810 Cable X	Kyocera	Decrypting Extraction using eMMC fault with Cable X. No soldering	3:49
17_SM-J100VPP	Samsung	Samsung Galaxy J1 SM-J100VPP EDL non-decrypting using test points	1:15
18_SM-S920L	Samsung	Samsung SM-S20L / SM-S530R4 – non-decrypting EDL using test points	1:18
19_SM-G360T1	Samsung	Samsung Galaxy Core Prime G360T1 / G360AZ / G360F / G360M / G360T. Using tweezers to create eMMC fault to create EDL mode. Non-decrypting extraction under Generic options in UFED 7.18	1:06
20_SM-S820L	Samsung	Samsung SM-S820L Galaxy Prevail / Core Prime / G360P / G360V. Using tweezers and test point to create EDL mode for non-decrypting EDL extraction in UFED 7.18	1:06
21_Y538 EDL Test Point Extraction	Huawei	Huawei Y538 Union. Using tweezers and test point for non-decrypting extraction in UFED 7.18.	1:04
22_A521L EDL - Test Point - Cable 523	Alcatel	Alcatel One Touch Pop Start LTE A521L / Pop Nova A520L. Non-decrypting extraction options using cable 523 / test points and Cable X	1:52
23_Google Pixel - Test Point EDL	Google	Google Pixel G-2PW4100. How to use hidden test point to create EDL mode using Cable X and voltage. Not supported for extraction.	1:00

Video	OEM	Description	Length
24_Huawei RIO-L01 EDL non-decrypting	Huawei	Huawei RIO-L01. Non-decrypting extraction using eMMC fault on underside of the logic board with tweezers. UFED 7.18	1:36
25_Samsung SM-A300FU EDL	Samsung	Samsung SM-A300FU. Non-decrypting extraction using Cable X to create eMMC fault.	1:42
26_ZTE N818S Looking for Test Points HB	ZTE	How to probe for unknown test points. Demo on ZTE Z818S using multimeter. Narration	4:58
27_SM-J510MN	Samsung	Samsung SM-J510MN. Non-decryping extraction using eMMC faults by shorting RST pad on eMMC using Cable X	1:08
28_DFU extract N818S	ZTE	ZTE N818S. How to research device not listed in UFED. Using DFU mode to perform non-decrypting extraction in Generic options in UFED.	2:33
29_non-decrypt extract A574BL from WFD screen	Alcatel	Alcatel A574BL. Non-decrypting extraction using Cable X and test point from the "Waiting for device" screen in UFED 7.18.	0:31
30_Decrypting A574BL - Test Points	Alcatel	Alcatel A574BL. Decrypting extraction using tweezers and test points to create EDL mode after charging indicator appears to avoid 2nd request for EDL mode. UFED 7.18	3:31
31_Probing E4610	Kyocera	Kyocera E4610. Start to finish investigation into location of unknown test point on device. Includes research, disassembly, eMMC faults with Cable X, use of multimeter, using pinout jig to probe and locate test point on opposite side of phone. Non-decrypting EDL extraction using hidden test point. UFED 7.18.	14:17
32_Z799VL Decrypting Extraction	ZTE	ZTE Z799VL. Decrypting extraction using tweezers and cable X to create EDL mode via test points on underside of logic board.	3:42
33_Z899VL Non-Decrypting Extraction	ZTE	ZTE Z899VL. Non-decrypting extraction using tweezers and Cable X to create EDL mode using a test point.	1:15
34_LG E960 Cable X	LG	Older LG E960. Demonstration of Cable X reliably creating EDL mode via eMMC faults on multiple tiny locations on logic board.	1:38pm
35_XT1644 - SIM Tool EDL Cable X	Motorola	Motorola XT1644 – using SIM removal tool as tool to create eMMC fault on Motorola Molex test point. Using Cable X	0:37
36_CLK EDL test original FCC	Kyocera	Kyocera E4610. Creating EDL mode by shorting CLK with Cable X	0.22
37_voltage test	Kyocera	Using multimeter to check for voltage to identify test points on Kyocera E4610	0.27
38_Cable X 1st Prototype Test Point and eMMC Fault Cable	Cable X – 1st prototype demo	Demo of first prototype Cable X to create eMMC faults using ground and voltage on several phones. Narrated	16:09
39_Damaged Samsung J3 USB Bypass	Samsung	Samsung J3 SM-J327P. Using USB bypass and pinning to extract badly damage encrypted device. UFED physical lock bypass for Samsung devices.	19:29

Video	OEM	Description	Length
40_LG EDL Removal Tool	LG	Using LG removal tool in UFED under LG EDL recovery tool. For devices stuck in EDL mode.	2:59
41_LGLS676 - LAF EDL	LG	LG LS676. Using EDL extraction under device profile in UFED for EDL extraction. EDL is created from LAF.	2:24
42_LGM150 - LAF EDL	LG	LG LGM150. Using EDL extraction under device profile in UFED for EDL extraction. EDL is created from LAF.	2:43
43_Smart ADB Extraction Motorola XT1765	Motorola	Smart ADB extraction in UFED. Swipe down and set "File Transfer" after phone reboots with cable 100. Charging mode after reboot will cause extraction to fail.	7:48
44 Huawei H1611 EDL Decrypting with Test Points	Huawei	H1611 Decrypting EDL extraction using test points	4:40
45 Desire 626s EDL	HTC	HTC Desire EDL extraction using eMMC Fault with Cable X	0:43
46 J415F EDL Extraction Fail	Samsung	Testing EDL on J415F. No programmer for this phone. Extraction fails.	0:55
47 Kyocera E6810 Decrypting	Kyocera	Most difficult phone in 3rd EDL webinar. Solder permanent wire to test point for decrypting EDL	8:25
48 Sony M4 E2306 EDL Non-decrypting	Sony	Non-decrypting EDL extraction using test points to trigger EDL	0:47
49_LG EDL Removal Tool	LG	Using the LG recovery tool in the UFED to remove permanent EDL mode created by improperly disconnecting phone	2:57
50 Koycera E5810 eMMC fault	Kyocera	Decrypting extraction using eMMC faults and UFED Touch2	6:58
51_LGLS676 – LAF EDL	LG	Decrypting EDL extraction with UFED's use of LAF to place the phone in EDL mode	2:23
52_LGM150 – LAF EDL	LG	Decrypting EDL extraction with UFED's use of LAF to place the phone in EDL mode	2:42
53 Motorola XT1527_1	Motorola	Motorola XT1527 Non-Decrypting EDL Extraction using test points	0:41
54 Huawei GR5 Decrypting EDL	Huawei	Huawei GR5 Decrypting EDL extraction using test points	4:12
55 Creating EDL Mode – 3rd Webinar	Various Devices	Various methods of creating EDL mode demonstrated. Tools and procedures demonstrated.	11:08
55 Huawei H1611	Huawei	Huawei H1611 Decrypting EDL – connect first, wait for battery charging, then create EDL with test point	4:02
56 Path to EDL Mode Webinar 3	Misc Devices	Video showing how to navigate the UFED for EDL extractions, common connection errors, and strategies	8:00
57 – 8 Extractions from 3rd EDL Webinar	Misc Devices	I selected 8 devices to demo EDL extractions and listed them from easiest to most difficult	30:14
58 ZTE Z852 Decrypting	ZTE	ZTE Z852 Decrypting extraction using test points	6:01
59 ZTE Z861BL Decrypting	ZTE	ZTE Z861BL Decrypting Extraction using test points	5:00

Video	OEM	Description	Length
60 Z899VL Decrypting	ZTE	ZTE Z899VL Decrypting Extraction using test points. Connect the device and wait for charging handshake, the use test points and power button to create EDL mode	6:43
61 ATT Axia - Damaged	ATT	What if your buttons are gone? ATT Axia QS5509A how to create EDL using volume and power button – without the buttons	7:29